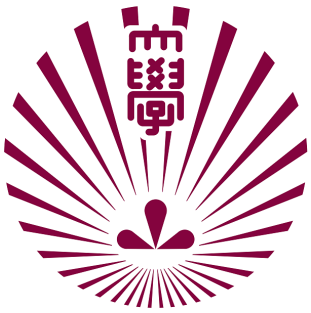
Lightweight Classification via Sub-Nyquist Compressive Measurements

ビリー, アレクサンダー, ドートン



Kyushu University

Graduate School of Information Science and Electrical Engineering
Department of Advanced Information Technology

Lightweight Classification via Sub-Nyquist Compressive Measurements

By

Billy Alexander Dawton

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Supervised by

Professor Yutaka Arakawa

August 2022

Billy Alexander Dawton
Lightweight Classification via Sub-Nyquist Compressive Measurements
Supervisor: Professor Yutaka Arakawa
Advisory Committee: Associate Professor Tsunenori Mine, Associate Professor Yutaka Jitsumatsu, and Associate Professor Shigemi Ishida

Kyushu University
Graduate School of Information Science and Electrical Engineering
Department of Advanced Information Technology

Abstract

As the foundation upon which the most important, exciting, and innovative technologies that pervade our everyday lives are built, it is no exaggeration to say that machine learning is the backbone of modern society. The benefits brought about both directly and indirectly by machine learning have, and continue to, revolutionise and change our lives for the better.

To deliver these benefits, however, the underlying systems require enormous amounts of data, which is costly both financially, and environmentally. Indeed, storing, accessing, and leveraging this data requires access to storage and computational infrastructure which is often expensive, sometimes prohibitively so. Furthermore, the acquisition of the data, as well as the training, testing, and deployment of the machine learning models themselves, requires substantial amounts of energy, which has a considerable negative impact on the environment.

The aim of our research is to tackle this problem at the source: by reducing the amount of samples required to perform classification, we look to lower the financial, computational, and energy-related costs incurred at every stage of the sensing-to-classification pipeline. To this end, we design a novel signal sensing and processing architecture, capable of acquiring signals at sub-Nyquist rates whilst simultaneously filtering them.

We begin by presenting our proposed sub-Nyquist sampling framework. Based on a modified random demodulator architecture, the system uses Markov-chains to create spectrally tailored spreading sequences to attenuate or amplify the frequency content of input signals during the acquisition process. We establish an initial design process by measuring and quantifying the effects of a range of bipolar spreading sequences on the frequency-domain content of a variety of input signals.

Next, we present two applications which make use of our proposed approach. The first application, C-AVDI, is a compressive measurement-based acoustic vehicle detection and identification system. The second application, C-HAR, is a compressive measurement-based human activity recognition system. The performance of both applications is gauged by comparing their respective classification accuracy and computational efficiency against currently available baseline systems. We find that in both cases, our proposed systems can be considered as lightweight alternatives to existing approaches in their respective fields.

Finally, we investigate the viability of our proposed approach as a lightweight sensing architecture by designing microcontroller (MCU) implementations of both C-AVDI and C-HAR. We find that the limited memory of MCU devices negatively impacts the performance of C-AVDI, but does not impact the performance of C-HAR.

The results obtained during the course of our research serve to prove the viability of our proposed compressive-measurement based sub-Nyquist architecture as a lightweight alternative to existing sensing-to-classification frameworks.

Acknowledgements

I would like to express my profound and heartfelt gratitude to my supervisor Professor Yutaka Arakawa for his constant help, support, and guidance over the last three years, without which this work would not have been possible. His relevant, pointed, and constructive advice helped shape and guide my research, ensuring it reached its full potential. I am truly grateful to him for making me feel welcome and at home in his laboratory, and for being so incredibly generous with his encouragement, support, and advice on starting a career in Japan.

I would also like to express my sincere and wholehearted thanks to Associate Professor Shigemi Ishida for his unrelenting support, precious advice, and kind mentorship during the course of my research. I am profoundly grateful for the time and energy he spent helping me conceptualise, conduct, and publish my research. I will remember fondly the times we spent discussing the intricacies of the English and Japanese languages. I owe him a great deal.

I am honoured to have Associate Professor Tsunenori Mine and Associate Professor Yutaka Jitsumatsu as advisory committee members. Their comments, advice, and feedback helped define and, when necessary, expand the scope of my work and helped ensure that it remained on track.

I would like to express my gratitude to Professor Akira Fukuda for accepting me as a research student in his laboratory, and to Professor Kenji Hisazumi for accepting to supervise me during that time.

I was lucky enough to be surrounded by fantastic colleagues and fellow researchers with whom I shared memorable times; in particular Mr. Jihed Makhlouf, Ms. Landy Rajaonarivo, and Mr. Ristu Saptono. I would also like to thank Mr. Yusuke Miyake and Mr. Toshihiko Sakai for taking the time every week to help me improve my Japanese, and Mr. Morgan Auffray for his

constant academic, professional, and emotional support over the last three years.

If it weren't for the valuable advice and assistance of the administrative staff, my life in Japan would have been very difficult. I would like to express my heartfelt thanks to Ms. Yasura Oiwa, Ms. Yuko Fukuda, Ms. Yukiko Hiranaka, and Ms. Chika Hamada for their help and support with everything.

Last but not least, I would like to thank my parents, my brothers, and my uncle Jules for their constant and unwavering support, inspirational guidance, and unconditional love. I would not have been able to achieve even a fraction of what I have until now if it weren't for them.

Contents

List of Figures

List of Tables

Acronyms

ADC	analogue-to-digital converter
ANN	artificial neural network
AVDI	acoustic vehicle detection and identification
C-AVDI	compressive measurement-based acoustic vehicle detection and identification
C-HAR	compressive measurement-based human activity recognition
CS	compressive sensing
CSP	compressive signal processing
CNN	convolutional neural network
DFT	discrete Fourier transform
DL	deep learning
DNN	deep neural network
DWT	discrete wavelet transform
FFT	fast Fourier transform
HAR	human activity recognition
ITS	intelligent transportation systems
k-NN	k-nearest neighbours
LPF	low-pass filter
MCU	microcontroller
ML	machine learning

MLP	multilayer perceptron
NLP	natural language processing
RD	random demodulator
RE	relative error
RF	random forest
RIP	restricted isometry property
SAVeD	sequential acoustic vehicle detector
SMBAS	stereo microphone-based AVDI system
STFT	short-time Fourier Transform
SVM	support vector machine
VDI	vehicle detection and identification

1 Introduction

1.1 Background

Machine learning (ML) is the keystone underpinning some of the most important technologies in modern society. Search engines, payment systems, self-driving cars, and numerous other technologies all depend on some kind of machine learning to operate and function. The advantages brought about by these technologies, however, come at a cost. There is a colossal amount of data used to create and operate these ML systems being created daily. In 2020, 59 zettabytes worth of data was created and analysed, and the prediction is that this will rise to 175 zettabytes by 2025 [1]. The implications of this are twofold: firstly, the storage, access, and use of these large amounts of data requires access to tools such as cloud storage and expensive computers, which makes both the use and the teaching of these technologies costly, and thus restricts their use in developing countries and economically disadvantaged areas. Secondly, the data centres used to store the data collected from sources and sensors across the globe require a significant amount of energy to function, as do the models which use the data.

This phenomenon is particularly apparent in the field of deep learning (DL), where the average number of calculations used in DL-related research have increased by a factor of 3×10^5 between 2012 and 2018 [2]. Indeed, there is currently in artificial intelligence- (AI) related research an overwhelming emphasis on obtaining the best possible performance (i.e. the best possible, accuracy, f-measure, precision or recall) in a given application. This becomes apparent when looking at recent research papers: the authors of [3], examined a set of 60 papers from three AI conferences (Association for Computational Linguistics (ACL) [1], Conference on Neural Information Processing

[1] https://acl2018.org

Systems (NeurIPS) [2], and Computer Vision and Pattern Recognition Conference (CVPR) [3]), noting whether the paper's main contribution to the field was an improvement in performance, efficiency, or both, and found that a significant majority of the papers reported an improvement in performance (90% of ACL papers, 80% of NeurIPS papers and 75% of CVPR papers). This phenomenon can also be observed when looking at the leaderboards of a popular AI competition website [4], where contestants are most commonly ranked by the performance of their submitted model, with no mention made of model size or efficiency.

While this heavy focus on performance in AI- and ML-related research does undoubtedly lead to the creation of new and cutting-edge techniques, approaches and applications, more often than not these improvements come at a significant environmental and financial cost. According to [4], in 2019 the process of training a single ultra-high-spec natural language processing (NLP) model (including parsing, semantic role labelling, model tuning and the successive inference stages), emitted more than twice as many kilograms of CO2 as the average American citizen does in a year (35592 kilograms vs 16400 kilograms).

Training cutting-edge text parsing algorithms costs in the order of several million dollars. Smaller companies are thus not able to compete with the biggest companies such as Google or Amazon. For instance, the cost of training GPT-3 [5], a cutting-edge, top-of-the-range NLP model, is estimated at $4.6 million [6]. MT-NLG, a new model jointly proposed by Microsoft and Nvidia, contains over 3 times as many parameters as GPT-3 [7]. These enormous costs make ultra high-end AI and ML completely inaccessible to anyone but the richest companies, severely limiting innovation and growth. This sentiment is echoed by Chris Manning, professor in ML at Stanford University who is quoted in [8] as saying: "I think it does cut down innovation, when we have only a handful of places where people can play with the innards of these models of that scale, that has to massively reduce the amount of creative exploration that happens."

[2] https://nips.cc/Conferences/2018
[3] http://cvpr2019.thecvf.com
[4] https://www.kaggle.com/competitions

This phenomenon is not just limited to large-scale NLP models, and is equally prevalent in smaller-scale models typically associated with energy- and cost efficiency such as embedded or long-timescale infrastructure monitoring applications. This "performance first" mentality is especially problematic in the field of medical devices: while one may expect the accuracy of a medical device to be the most important metric, studies have found that the biggest barrier to the effectiveness of wearable medical sensors is the inconsistent use of the devices by the patients. Indeed, while a high-performance device may perform better, these devices are usually more power-hungry and thus require consistent charging, which has a significant negative impact on the long-term use and adoption of the device [9].

Typically, the data used in ML applications is a digital-domain representation of an analogue-domain signal. The process in which a continuous signal is transformed into its discrete representation is known as analogue-to-digital conversion. This procedure is governed at the most fundamental level by the Shannon-Nyquist sampling theorem, which states that to accurately capture and digitise an analogue signal, it is necessary to sample it at a rate at least twice as high as the signal's highest frequency. This ensures that enough information is acquired for the signal to be reconstructed properly, and that aliasing is avoided. This rule governs all devices that perform analogue-to-digital conversion, and sets a minimum performance rate for the hardware involved in the signal acquisition. Due to its ubiquitous nature, the lower limit on sample rates mandated by this theorem has had a direct and significant effect on the raw amount of data being created, transmitted, and processed.

In response to these issues, the authors of [3] propose a set of metrics which can be used to measure and gauge the overall efficiency of any given ML architecture. Of particular interest are the following three:

- *Energy usage:* a system's energy usage is correlated with its carbon emission levels. This metric is of particular concern when dealing with large-scale GPU-powered networks with large carbon footprints, but is also important to consider when designing low-cost, low-complexity MCU-based systems to ensure that they perform as intended while staying under their allocated energy budget.

- *Running time:* the running time of a supervised learning model is a clear indicator of system efficiency, as for two models running on the same hardware, the most efficient one will run faster. This metric is particularly useful when prototyping different approaches in clean, reproducible, software environments (such as Docker containers).
- *Model parameters and size:* The number of parameters used by a model is proportional to its size. As smaller models can be deployed on less power-hungry, smaller form factor hardware, it can thus be said that a model's size is proportional to its energy consumption. It is also important to consider a model's size when deploying models in remote or edge situations in which system and battery autonomy are key factors.

Taking the above criteria into account when evaluating models rather than focussing exclusively on performance, can help guide the creation of new, energy and cost conscious AI and ML architectures.

1.2 Research Objectives and Contributions

The objective of this research is to create a novel simultaneous signal sensing and processing architecture which directly obtains relevant information from an analogue input signal, in effect compressing and filtering whilst sampling, and using it in tandem with ML in a variety of classification-focussed applications. We do so by exploiting the underlying structure of the information or signal we are looking to acquire to sample it at sub-Nyquist rates using compressive sensing (CS). By reducing the raw number of samples required to perform classification, we reduce the associated cost and complexity at each stage of the sampling-to-classification pipeline. In this thesis, we use the term "lightweight" to mean "low-cost, low-complexity", and refer to the samples obtained during the CS process as "compressive measurements", thus we refer to our proposed approach as: "Lightweight Classification via Sub-Nyquist Compressive Measurements".

We consider a sensing architecture lightweight if it can feasibly be deployed on MCU devices. This introduces a set of considerations to the system design,

as it is not only the limited MCU clock speed that needs to be taken into account, but also the limited ROM and RAM. While certain systems such as the one described in [10] attempt to mitigate this problem by transferring the data obtained by an MCU-based sensor node to an edge node or the cloud for processing and inference, this is not always feasible. Indeed, in certain security- or privacy-critical applications, transferring data from the sensor node is a potential security risk, as it can be intercepted by a malicious actor. Furthermore, not only does the transmission process in itself consume energy, but creating, designing, and deploying the appropriate receiving architecture increases the overall cost of the sensing system. Therefore, for our approach to be considered truly lightweight, it is crucial that any potential applications based on our proposed architecture can be deployed on MCUs.

In our approach, we seek to improve the efficiency of the sensing-to-classification process at the source. While there are a lot of potential inefficiencies when collecting and using data, the biggest inefficiency is at the very start of the process: we acquire our data by sampling, compressing, and then discarding any unwanted or redundant information, which as a process is inherently inefficient. By reducing the number of measurements required by the classification architecture, we are looking to lower the cost, complexity, and energy requirements associated with ML-related classification, helping to reduce the environmental impact of these technologies while making them as accessible as possible.

While the concept of lightweight sensing-to-classification pipelines is not novel, as evidenced by approaches such as those in [11] and [12], the concept of compressing while sampling remains largely unexplored. Recently, however, a similar approach to the one put forward in this thesis was presented in [13], in which the authors designed a low-power compressive sensing-inspired sub-Nyquist sensing device, capable of detecting electrocardiogram (ECG) anomalies. By using Non-Uniform Wavelet Sampling (NUWS) to obtain a set of features directly from the ECG signal, the authors are able to detect occurrences of cardiac arrhythmia with very high accuracy and significantly reduced power consumption. The similar goals and timing of publication of both approaches highlights the growing importance of lightweight classification in the field of ML. It is important to note, however, that despite the similarities, there are considerable differences that exist

between the two proposed architectures: in our proposed approach we are looking to perform compressive-measurement based multiclass classification with the aim of reducing the lower-bound on the number of samples needed for classification, rather than anomaly detection using the samples obtained by NUWS as features with the aim of reducing power consumption. In addition to this, our proposed system is capable of simultaneously filtering and sub-Nyquist sampling of a given input signal, while the system presented in [13] is not.

In this thesis, we are looking to address one of the fundamental limits of traditional sensing architectures: the sampling rate. In particular, we are looking to develop a powerful, application-agnostic alternative to existing classification pipelines and sensing architectures. To this end, we present two applications which make use of our proposed lightweight compressive measurement based-classification architecture. The first is C-AVDI, a compressive measurement-based acoustic vehicle detection and identification system, which performs multiclass classification on vehicle sounds. The installation and operation costs associated with acoustic vehicle detection and identification (AVDI) are usually low, but the preprocessing of signals is often computationally complex, and the typically high sample rates associated with audio signals make the acquisition and storage of longer signals costly in terms of memory. To address this issue, our proposed approach simultaneously filters and samples at sub-Nyquist rates, removing the need for a dedicated preprocessing stage, and greatly reducing the amount of samples required for classification. C-AVDI obtains an accuracy comparable to competing systems with a sample rate 16 times slower than the Nyquist rate. The second is C-HAR, a compressive measurement-based human activity recognition system, which performs multiclass classification on smartwatch sensor readings. There are two factors which have a significant impact on the usage and adoption rates of smartwatch-based human activity recognition (HAR) devices: battery life and data privacy. By simultaneously filtering and sampling input signals at sub-Nyquist rates, battery life can be improved by removing the need for a complex dedicated filtering stage, while reducing the minimum number of required samples allows for a greater amount of user data to be stored, processed and classified on-device, helping safeguard user privacy. C-HAR obtains an accuracy comparable to competing systems with a sample rate 4 times slower than the Nyquist rate.

1.3 Thesis Outline

We begin in Chapter 2 by introducing the fundamental concepts upon which the work in this thesis is built, covering different sampling approaches, architectures and applications.

We then examine some existing baseline approaches in Chapter 3. First we present a stereo microphone-based acoustic vehicle detection and identification system (SMBAS), our initial approach to performing AVDI. Vehicle detection is performed using a soundmap-based method based on the work presented in [14], and vehicle classification is performed using short-time Fourier transform (STFT) features extracted from the combined time-shifted sum of the two microphone signals. Despite its hybrid soundmap- and STFT-based approach, SMBAS is representative of more traditional Nyquist rate sample-then-compress AVDI approaches, making it a suitable benchmark against which to compare our subsequently proposed lightweight AVDI system. We then briefly explore a selection of existing HAR approaches, focussing in particular on smartwatch-based systems.

In Chapter 4, we present a sub-Nyquist sampling strategy which simultaneously filters and samples input signals for use in classifier-based systems. This involves running a battery of reconstruction tests on a set of input signals using a range of Markov chain-generated bipolar spreading sequences. By observing the effects the different spreading sequences have on signal reconstruction, we are able to iteratively determine the optimal spreading sequence generation parameters for any given situation.

Next, we explore two applications in which we apply the simultaneous sub-Nyquist sampling and filtering signal acquisition strategy. In both cases, classification is performed using features extracted directly from the sub-Nyquist compressive measurements.

The first application, presented in Chapter 5, is a compressive measurement-based acoustic vehicle detection and identification (C-AVDI) architecture, in which passing vehicles are detected and identified using sounds obtained by acoustic sensors placed on the side of the road. The second application, presented in Chapter 6, is a compressive measurement-based human activity recognition (C-HAR) architecture, in which a selection of daily activities

are identified from a smartwatch's three-axis gyroscope and accelerometer readings. The performance of both applications is gauged by comparing their classification accuracy and computational efficiency against the benchmark systems.

In Chapter 7, we investigate the viability of our proposed approach as a lightweight sensing architecture by designing MCU implementations of both the C-AVDI and C-HAR systems.

Finally, in Chapter 8, we discuss and reflect upon the results obtained during our research, and outline relevant and promising directions for future work.

2 Fundamental Concepts

2.1 Nyquist Rate Sampling

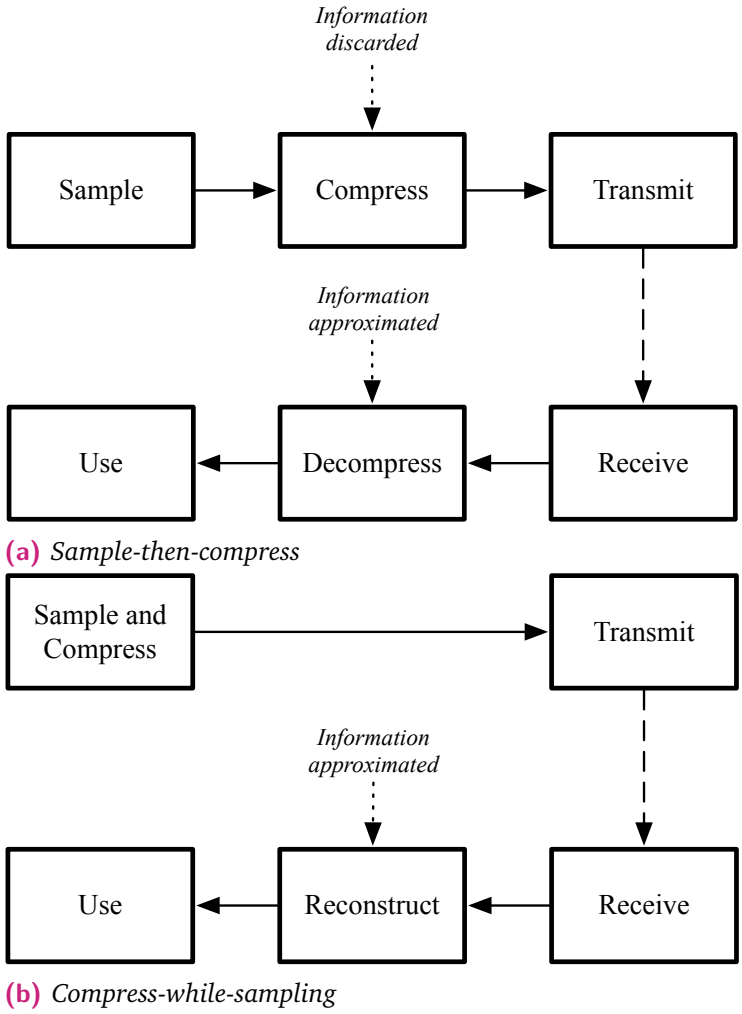
The process in which a continuous-time, continuous-amplitude signal is discretised for use in digital applications is called analogue-to-digital conversion. This is done using a device called an analogue-to-digital converter (ADC) which assigns a numerical value to the input signal's amplitude at pre-determined sampling intervals, discretising the original signal in both time and amplitude. To reconstruct the signal effectively, it must be sampled at twice the highest frequency of the original signal; this rate is called the Nyquist rate and is the cornerstone of modern-day signal processing. For a continuous time-domain signal $x(t)$ whose highest frequency component is denoted as $W/2$ Hz, the Nyquist rate can be defined as:

$$f_s = W \text{ Hz} \tag{2.1}$$

As ubiquitous as this technique is, it has limitations when confronted with wideband analogue signals due to the inability of current hardware to sample at a fast enough rate. This impracticality is compounded by the inherent inefficiency of analogue-to-digital conversion, as while sampling at the Nyquist rate ensures that all the information contained in a signal is recovered, often not all the information is needed. The use of compression algorithms post-sampling removes redundant information present in the input signal, reducing the size of the signal's digital representation in memory, however this practice is wasteful as we are discarding information the ADC was designed to acquire. The benefits of designing a sampling paradigm capable of obtaining exclusively the relevant information would enable us to reduce the required sampling rate, replacing the traditional *sample-then-compress* paradigm with

an alternative and improved *compress-while-sampling* paradigm [15], both of which are shown in Figure 2.1.

In certain sensing applications, the signals being acquired present a sparse signal structure; in other words, a given signal x contains only K non-zero elements out of N total elements, with $K \ll N$. In such cases, it is possible to take advantage of the signal's sparsity and sample it at a sub-Nyquist rate using a signal processing technique called compressive sensing (CS).



(a) *Sample-then-compress*

(b) *Compress-while-sampling*

Fig. 2.1: The *sample-then-compress* and *compress-while-sampling* paradigms.

2.2 Compressive Sensing

2.2.1 Basic Concepts

First presented in [15] and [16], compressive sensing (CS) is a method to efficiently sample sparse or compressible signals (a signal is called sparse if it contains only a few nonzero components compared to its total length, and is called compressible if it contains many nonzero components, but only a few of significant magnitude), at sub-Nyquist rates enabling us to directly acquire the information of interest in a given signal, rather than sample the signal at the Nyquist rate and subsequently discard unwanted information.

Signals themselves are typically represented mathematically as a continuous range of values as a function of time, or as a linear combination (i.e. a weighted sum) of sinusoids, from which we can extract the amplitude coefficient and frequency information of each sinusoid in the linear combination to obtain the frequency domain representation of a signal. This concept can be extrapolated further, and for a continuous signal in the time-domain, we can also express signals as a linear combination of basis vectors and discrete coefficients over a given time period. Thus, let us begin by defining a real-valued sparse signal $x(t)$ of length T_s whose highest frequency component is $W/2$ Hz. Let $N = WT_s$ be the dimensionality of $x(t)$. This signal can be expressed as a combination of discrete coefficients $\alpha \in \mathbb{C}^N$ and vectors ψ_n that form the columns of an orthonormal basis matrix $\mathbf{\Psi} \in \mathbb{C}^{N \times N}$ for a given time window:

$$x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t) \ , \ t \in [0, T_s) \tag{2.2}$$

with the coefficients computed as $\alpha_n = \langle x, \psi_n \rangle$, and whose vector matrix form is:

$$x = \mathbf{\Psi} \alpha \tag{2.3}$$

with $\mathbf{\Psi}_{m,n} = \psi_n \left(\frac{m}{N} T_s\right)$, from which we can obtain our signal's discrete-time notation as an N-dimensional vector:

$$x_m = x\left(\frac{m}{N} T_s\right) \tag{2.4}$$

More often than not, $x(t)$ itself is not sparse and instead has a sparse representation in $\mathbf{\Psi}$. Given prior knowledge of $\mathbf{\Psi}$, we need only obtain the information contained in the sparse coefficient vector α to be able to reconstruct the original signal. This information is obtained by drawing a set of compressive samples $\xi \in \mathbb{R}^M$ from the original signal, where $M \ll N$.

The CS acquisition process can be described mathematically as:

$$\xi = \mathbf{\Phi} x \tag{2.5}$$

where $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ represents the consecutive sampling and operations performed on x.

Additionally, we define the reconstruction matrix as: $\mathbf{\Theta} \in \mathbb{C}^{M \times N} = \mathbf{\Phi\Psi}$.

From (2.2) and (2.5):

$$\xi = \mathbf{\Phi\Psi}\alpha \tag{2.6}$$

$$= \mathbf{\Theta}\alpha \tag{2.7}$$

The sparse coefficient vector and thus the original signal can be recovered from the compressive measurements by solving the l_1 minimization problem:

$$\hat{\alpha} = \arg\min_{\alpha} \|\alpha\|_1 \quad \text{such that} \quad \xi = \mathbf{\Theta}\alpha \tag{2.8}$$

where $\hat{\alpha}$ is the estimated coefficient vector and $\|\alpha\|_1$ is the l_1 norm (sum of the absolute vector values) of α. In practical terms, this can be achieved by

using a recovery algorithm; either a convex relaxation-based one such as basis pursuit (BP) [17] or a greedy one such as orthogonal matching pursuit (OMP) [18].

For complete recovery of x, it is necessary to design $\mathbf{\Theta}$ in such a way that it satisfies the incoherence and RIP (restricted isometry property) conditions outlined in [19], thereby ensuring that all the relevant information present in x is preserved in the measurements ξ. In [20], it is stated that a matrix such as $\mathbf{\Theta}$ in (2.7) satisfies these conditions with overwhelming probability if $\mathbf{\Psi}$ is an orthonormal basis, $\mathbf{\Phi}$ is drawn randomly from a suitable distribution such as the Gaussian distribution, and if the number of measurements M is higher than a lower bound defined as:

$$M \geq SK \log \frac{N}{K} \tag{2.9}$$

where K is the sparsity level of the input signal and S is a positive constant [21].

2.2.2 The Random Demodulator

The initial theoretical work on CS as described above only considers discrete signals, however our proposed systems look to obtain continuous-time signals. First presented in [22] and further expanded upon in [23] and most notably in [24], the random demodulator (RD) is a signal acquisition architecture that enables us to perform CS on sparse or compressible continuous-time signals. Compared to more sophisticated architectures such as the modulated wideband converter (MWC) [25] and quadrature analog-to-information converter (QAIC) [26], the RD is more straightforward to design and cheaper to implement as it is single channel and only requires a single ADC. Furthermore, the RD is particularly suitable for our applications as it is designed to acquire sparse single-band multitone signals, unlike the MWC for instance, which is a multichannel architecture designed to acquire sparse multiband signals. More details on different CS acquisition strategies can be found in [21], and [27] provides an in-depth comparison of the RD and MWC systems in particular.

We can give an intuitive description of the RD's operation as follows: rather than acquiring a signal through traditional Nyquist sampling, the RD first demodulates the signal by multiplying it with a white noise-like pseudorandom sequence, spreading the signal's frequency content across the entirety of the spectrum. The resulting signal is then low-passed before being sampled at a sub-Nyquist rate. If required, the original signal can be recovered from the sub-Nyquist compressive samples through l_1 minimization as explained in Section 2.2.1.

Let us describe the operation of the RD more formally. An analogue signal as described in (2.2) is combined with a pseudorandom bipolar sequence of unitary amplitude, defined as:

$$PRS(t) = \epsilon_n, \;\; t \in \left[\frac{n}{C}, \frac{n+1}{C}\right), \;\; n = 0, 1, ..., N-1 \tag{2.10}$$

where ϵ_n is a Rademacher sequence that switches between values $\{-1, 1\}$ at a rate $C = W$.

The combined signal $x(t)PRS(t)$ is passed through an anti-aliasing LPF $h(t)$ of bandwidth $R/2$ and sampled at a rate $R < W$ to obtain linear compressive samples $\xi[m]$. This procedure can be expressed as a multiplication followed by a convolution in the time domain:

$$\xi[m] = \int_{-\infty}^{\infty} x(\tau)PRS(\tau)h(t-\tau)\,d\tau\Big|_{t=mR} \tag{2.11}$$

$$= \sum_{n=1}^{N} \alpha_n \int_{-\infty}^{\infty} \psi_n(\tau)PRS(\tau)h(mR-\tau)\,d\tau \tag{2.12}$$

From which we obtain an expression for $\mathbf{\Theta}$, whose entries are defined as $\theta_{m,n}$ for row m and column n:

$$\theta_{m,n} = \int_{-\infty}^{\infty} \psi_n(\tau)PRS(\tau)h(mR-\tau)\,d\tau \tag{2.13}$$

where $\mathbf{\Theta}$ is a combination of the matrix $\mathbf{\Phi}$ which represents the sequence of operations mapping the input signal x to the compressive measurements ξ,

and of the orthonormal basis matrix $\mathbf{\Psi}$. It is shown in [24] that $\mathbf{\Theta}$ satisfies the previously outlined RIP conditions, as long as the number of measurements matches or exceeds M as described previously in Equation 2.9

Finally, it is necessary to ensure that in any given application, the input signals to the RD are sparse or compressible in the domain defined by $\mathbf{\Psi}$. Signals are only very rarely perfectly sparse and are much more likely to be compressible, that is, the magnitudes of the nonzero coefficients present in the signal decay following a power law distribution. This is defined in [28] as:

$$|\alpha_n| \leq Pn^{-q} \text{ for } n \in \{1, 2, ..., N\} \tag{2.14}$$

where P and q are constants.

2.3 Intelligent Transportation Systems

The increasing development of information and communication technology in recent years has led to similar advances in the field of intelligent transportation systems (ITS). A growing number of ITS applications such as navigation, traffic dependent guidance and auto-cruise systems have been proposed and realized with the aim of improving road traffic safety, efficiency, convenience, and reliability.

2.3.1 Vehicle Detection and Identification

The process of vehicle detection and identification (VDI) plays a key role in delivering the many economical, societal, and environmental benefits of ITS technology. A host of applications rely on VDI to operate: traffic flow control [29], to reduce congestion, pollution, and fuel consumption by effectively routing traffic around road networks; road safety management and collision avoidance [30], to reduce the number of fatal and near-fatal accidents; and infrastructure management [31], to monitor the usage and condition of various transport networks, helping repair and maintain them

in the most optimal manner. The ubiquity and variety of these applications serve to highlight the importance of VDI as a part of ITS as a whole.

The methods used to implement VDI are varied and numerous, and can be split into two broad categories. The first category includes intrusive methods, based on video cameras [32], LIDAR [33], radar [34], inductive loop coils [35], and magnetic sensors [36], which present good detection and identification performance, but require the use of task-specific hardware which typically incurs substantial installation and running costs. The second category includes non-intrusive methods, which use technologies such as on Wi-Fi channel state information (CSI) [37] and acoustic sensors [38], which are cheaper to install and operate, and in some cases able to function by piggybacking off pre-existing infrastructure.

2.3.2 Acoustic Vehicle Detection and Identification

Acoustic vehicle detection and identification (AVDI) is a subcategory of VDI which makes use of acoustic sensors to obtain information from passing vehicles. The non-intrusive nature of acoustic sensors make AVDI an easy-to-install and low-cost alternative to more expensive intrusive VDI systems. Moreover, the small form factor of acoustic sensors allows them to be deployed quickly and easily in remote locations.

There is a wide range of existing AVDI implementations, each of which present different advantages and disadvantages, making them suited to different roles and situations. From a high-level perspective, the most commonly used approach is *sample* $\rightarrow$ *process* $\rightarrow$ *classify*, typically including feature extraction during the *process* stage. There are a substantial number of systems that follow this general approach, but differ significantly in their implementations: input audio signals in mono [39] or stereo [40]; MFCC [41], DFT [42], or DWT [43] as domain transforms; detection and identification performed using supervised [44] or unsupervised learning [45]. These systems each present a different trade-off between cost, complexity, and performance, and the choice of which system to use depends on the application.

There are also AVDI systems which use different, more heuristic, high-level approaches, most notably sound map-based systems such as [14] which uses

RANSAC, and [46] which uses dynamic time warping to identify passing vehicles, along with their direction of travel, but do not perform vehicle type classification.

2.3.3 Lightweight AVDI

The inherently lightweight nature of AVDI makes it particularly suitable for use in low-power MCU-based systems deployed in remote or edge ITS environments. This is typically done by focusing on two key aspects: battery life and system autonomy. Extending the system's battery life ensures that it can operate reliably for extended periods of time, while also reducing the frequency at which the system needs to be taken offline to be recharged or replaced. Optimising system autonomy allows the system to store and process as much data as possible on the device itself, sending no, or very limited amount of information back from the lightweight sensing device to a central processing node.

In practice, this can be achieved in a number of different ways, for instance the authors of [47] propose an ultra low-power vehicle detection system which uses the discrete Wavelet transform (DWT) and logistic regression, while the authors of [11] present an environment monitoring system which uses FFTs and a hidden Markov model (HMM).

2.4 Human Activity Recognition

Human activity recognition (HAR) can be defined as the task of identifying the actions, activities, and movements of human subjects, and plays a crucial role in a wide range of applications such as healthcare monitoring [48], assisted living [49], and both cyber and physical security [50, 51].

The sensors used to perform HAR can broadly be separated into two categories. The first category includes remote sensors such as video [52], radar [53], or Wi-Fi CSI channel information [54], and are more commonly used in applications where multiple subjects are being monitored simultaneously, such as in public spaces. The second category includes wearable sensors

placed directly on the subject in the form of either custom hardware [55], or using sensors present in everyday objects like smartphones [56], smartwatches [57], most often used to monitor the movements and activities of individual subjects.

While wearable sensors used in HAR initially consisted of purpose-built devices, improvements in technology have led to the widespread adoption and use of sensor-equipped smartphones and smartwatches, causing a rise in both the number of people carrying sensors and the period of time the sensors are carried for, significantly increasing the viability of individual subject-oriented HAR. Of the two devices, the smartwatch can be considered the most important as it is always placed in the same place, the wrist (a part of human anatomy with multiple axes of movement) and whose movements vary significantly depending on the activities performed by the wearer. In contrast, the smartphone is limited to few degrees of movement due to usually being placed in a pocket, on a table, or in a bag. It follows that studies on individual subject-oriented HAR tend to use data collected from smartwatches or a combination of both smartwatch and smartphone data, paired with supervised learning to detect and identify a subject's activities.

2.4.1 Lightweight HAR

Battery life and system autonomy are crucial factors in many HAR systems, especially those found in safety-critical applications where usage and adoption need to be as consistent and as uninterrupted as possible. A survey conducted by [9] as part of a study on the large-scale deployment of multiple wearable sensors in Parkinson's disease found that 27% of participants stopped using their wearable sensing devices due to difficulties ensuring they remained charged. Lightweight approaches typically focus on minimising model size and power consumption; a lightweight example of the first type of system can be seen in [58], which uses a simple deep neural network-(DNN) based framework capable of both offline and online learning, and a lightweight example of the second type of system can be seen in [13] which uses a compressive ultra low-power wavelet-based system to perform anomaly detection.

3 Baseline Approaches

3.1 Introduction

Baseline approaches are an important factor to consider during the design process of a new system, as they are an effective way to gauge the performance and viability of any proposed architectures. In this chapter, we focus primarily on the acoustic vehicle detection and identification (AVDI) system we designed in [40] as it is simultaneously an important first step in the creation of our compressive measurement-based acoustic vehicle detection and identification (C-AVDI) system, while also being sufficiently different to the final system in terms of operation to act as a suitable benchmark. We also briefly cover some existing baseline human activity recognition (HAR) approaches, as given the popularity of the field, there is a large body of work to draw from. For the purpose of making any comparisons as accurate as possible, we focus in particular on smartwatch-based systems.

3.2 SMBAS: Stereo Microphone-Based AVDI System

3.2.1 Background and Related Work

The detection and identification of vehicles is of paramount importance in a wide variety of intelligent transportation system (ITS) applications, and several methods have already been put forward for this purpose. The vast majority of approaches, however, only focus on vehicle detection and no consideration is given to the identification, or classification, of vehicle

type. Recently, there has been an increasing demand for systems capable of performing vehicle identification as well as vehicle detection.

There is a wide variety of existing vehicle identification approaches, which can be split into a number of different categories.

- *Camera-based systems:* the authors of [32] propose a system capable of automatically recognising a vehicle's number plate, and thus its registered information, from a single image using edge statistics. This method requires the use of a high-performance computer for analysis, and the installation of a camera in front of the vehicle passing point to achieve high accuracy. In [59] the authors put forward a classification method using vehicle length: by taking the background difference information from images taken from roadside surveillance cameras, the authors are able to obtain the passing time, and thus the length of passing vehicles which is then used to determine vehicle type. Whilst this method is effective for detecting long vehicles such as trucks or buses, it is not suited for shorter ones like cars or motorbikes. In addition, accuracy performance suffers in rainy and foggy situations.

- *Electronic toll collection- (ETC) based systems:* in ETC-based methods, the vehicle type is identified by the registration information contained in the ETC onboard equipment. Whilst ETC systems enjoy common and widespread use on motorways in Japan and around the world, the high installation and maintenance costs of the infrastructure make it difficult for them to be installed on standard roads for the sole purpose of vehicle detection and identification.

- *Acoustic sensor-based systems:* the authors of [44] and [60] propose approaches using frequency domain features in tandem with supervised learning setups such as support vector machine (SVM) and k-nearest neighbour (k-NN) classifiers. The authors of [61] developed a method for estimating a vehicle's type based on the envelope shape of its emitted acoustic signal in the frequency domain. As each vehicle has a unique frequency spectrum shape, the system is able to accurately distinguish individual vehicles from one another. However, the inherent uniqueness of each frequency spectrum shape makes it impossible for

the system to classify a passing vehicle's type (i.e. to determine its class label).

We are looking to create a lightweight, yet accurate, vehicle detection and identification (VDI) system. We previously mentioned in Section 2.3.2, that acoustic vehicle detection and identification (AVDI) can act as an inexpensive, low-complexity VDI method. Thus, we present SMBAS, a stereo microphone-based AVDI system capable of detecting and identifying passing vehicles as a low-cost, easy-to-implement alternative to the VDI systems outlined above.

3.2.2 System Overview

The intuition behind SMBAS is as follows: a stereo microphone pair is placed on the side of the road to track a vehicle's position relative to both microphones, as the vehicle passes in front of each microphone successively, its sound is recorded and the time difference between both microphones is calculated from which we can obtain the direction and speed of the passing vehicle. The passing vehicle's sound signature is then emphasized by aligning and superimposing the signals obtained by each microphone.

Figure 3.1 shows an example of a vehicle moving from right to left: the left channel microphone is located further from the vehicle than the right channel microphone. The arrival time between the sound emanating from the vehicle and the left microphone is larger than the arrival time between the sound emanating from the vehicle and the right microphone. The difference between the arrival times is Δt.

By shifting the left channel sound by $-\Delta t$ and adding it to the right channel sound, we obtain our combined emphasized sound. We set $s_L(t)$, $s_R(t)$, as the left and right channel audio signals respectively and $s_{\text{emph}}(t)$ as the emphasized signal:

$$s_{\text{emph}}(t) = s_R(t) + s_L(t + \Delta t). \tag{3.1}$$

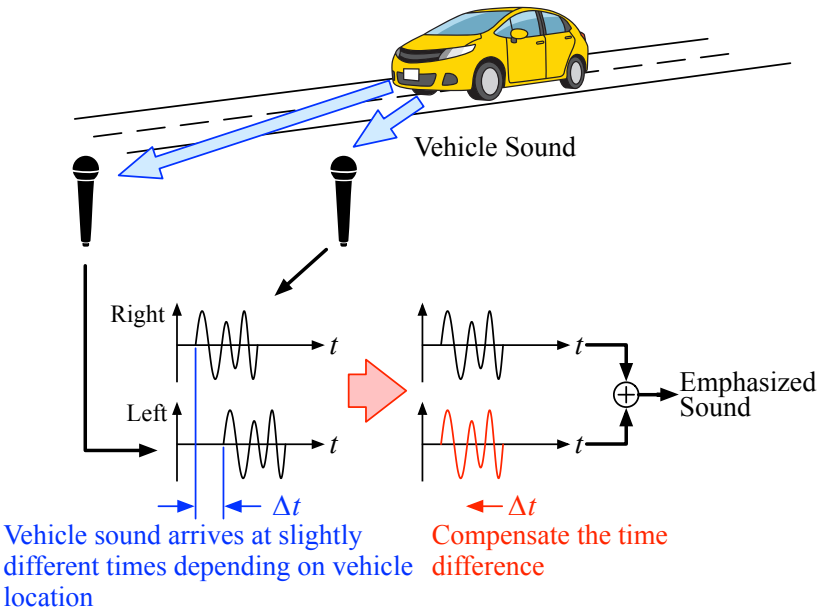


Fig. 3.1: A passing vehicle's sound signature is emphasized by aligning and superimposing the signals obtained by the microphones.

The vehicle type is estimated from this emphasized signal using supervised learning methods. Since the vehicle is assumed to travel continuously along the road, Δt changes with time and is thus a function of time t:

$$s_{\text{emph}}(t) = s_R(t) + s_L\left[t + \Delta t(t)\right]. \tag{3.2}$$

This concept, which underpins the operation of our proposed system, is described as a continuous time process, but its actual implementation is done in the digital domain.

Figure 3.2 shows a system overview of SMBAS consisting of the following components: a *Sound Retrieval* block, a *Vehicle Detection* block, an *Emphasis Synthesizer* block, and a *Vehicle Type Classification* block.

The *Sound Retrieval* and *Vehicle Detection* blocks listen for sounds and analyse them to detect passing vehicles; if a vehicle is detected, then the two blocks will both acquire the vehicle passing time and the reception time difference Δt. The detection block is designed using the sequential acoustic vehicle detection (SAVeD) method established in previous research [14]. Using the acquired Δt, the sound signals acquired by the left and right microphones are superimposed in the *Emphasis Synthesizer* block to enhance the vehicle sound

in the direction of travel. Frequency domain feature values are extracted from this emphasized audio signal, and the vehicle is identified using supervised learning in the *Vehicle Type Classification* block.

The workings of each block are explained in the following sections.

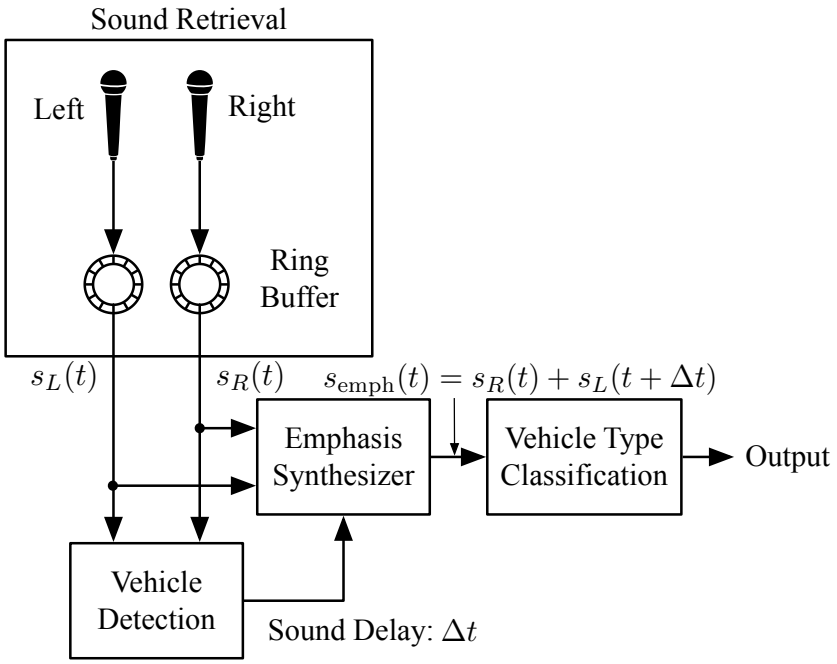


Fig. 3.2: System overview: SMBAS consists of a *Sound Retrieval*, *Vehicle Detection*, *Emphasis Synthesizer*, and *Vehicle Type Classification* block.

3.2.3 Sound Retrieval Block

The *Sound Retrieval* block is composed of a stereo microphone pair. Figure 3.3 shows the experimental microphone layout: the two microphones M_1 and M_2 are installed at a distance D from each other and a distance L from the road. As the distances d_1 and d_2 from the vehicle to each microphone change over time, so does the time delay between a sound being emitted by a vehicle and it reaching both microphones. The audio signals acquired by both microphones are temporarily held in ring buffers for later use, as the time difference between the signals is used in both the detection and classification processes.

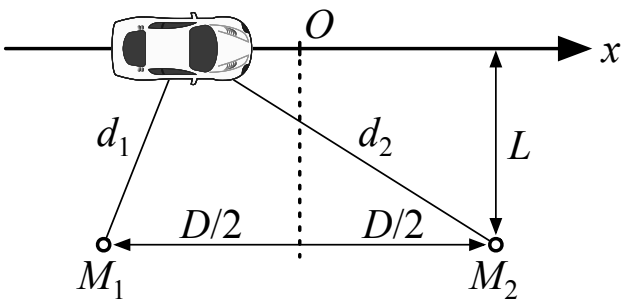


Fig. 3.3: Microphone positioning and setup.

3.2.4 Vehicle Detection Block

The *Vehicle Detection* block uses the audio signals stored in the ring buffers to detect passing vehicles by drawing a soundmap. We write the audio signals received by the two microphones as $s_1(t)$ and $s_2(t)$, and the cross-correlation function $R(t)$ as:

$$R(t) = \int s_1(t)\, s_2(t+\tau)\, d\tau. \tag{3.3}$$

If the two microphones receive a signal with a time difference of Δt such as: $s_1(t) = s_2(t + \Delta t)$, then $R(t)$ reaches its maximum value at $t = \Delta t$. The time difference Δt can then be estimated by looking for the peak of $R(t)$. The actual value of Δt is calculated using GCC-PHAT (Generalized Cross-Correlation Phase Transform) which calculates the time difference in the frequency domain.

Additionally, on Figure 3.3 we can see that the difference in reception time (or sound delay) Δt between microphones M_1 and M_2 is proportional to the distance between the sound source and each microphone respectively. We set the initial passing time of a vehicle in front of the centre of the microphones as $t = t_0$.

We thus also derive $\Delta t(t)$, which is a function of time, in the following manner:

$$\begin{aligned}\Delta t(t) = & \frac{d_1 - d_2}{c} \\ = & \frac{1}{c}\left\{\sqrt{\left[v(t-t_0)+\frac{D}{2}\right]^2 + L^2}\right. \\ & \left. -\sqrt{\left[v(t-t_0)-\frac{D}{2}\right]^2 + L^2}\right\}, \end{aligned} \tag{3.4}$$

where c is the speed of sound.

From (3.4), we can see that as a vehicle passes in front of a microphone with a constant speed v, an S-shaped curve is drawn on the soundmap. The *Vehicle Detection* block works by detecting this curve using a random sample consensus (RANSAC) robust estimation algorithm [62]. In our case, the unknown parameters in (3.4) are the speed v and the initial passing time t_0; these are estimated by fitting (3.4) to a "high likelihood" point cloud on the soundmap.

Figure 3.4 shows an example of vehicle detection using a soundmap and RANSAC, with the blue dots indicating the sound delay at each time t, and the orange line the result of the RANSAC fitting process. The red points were judged as being of "high likelihood" during the fitting process. For each detected vehicle, the *Vehicle Detection* block calculates and outputs the speed v and the passing time t_0 to the *Emphasis Synthesizer* block.

3.2.5 Emphasis Synthesizer Block

The *Emphasis Synthesizer* block begins by calculating the passing sound time difference Δt at each time t using the speed v and the initial passing time t_0 for each vehicle detected by the *Vehicle Detection* block. Using this information, the *Emphasis Synthesizer* block shifts the received sound signal at one of the microphone channels in time and adds the sounds of both channels together, creating an emphasized sound.

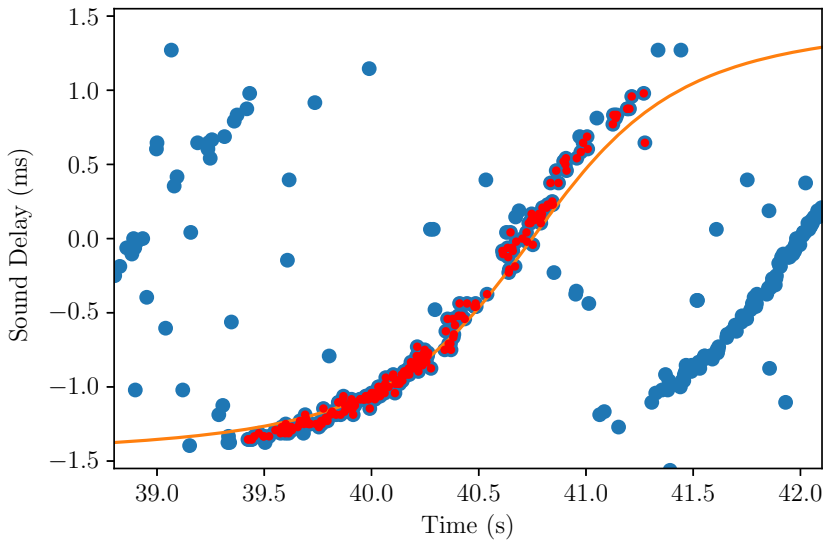


Fig. 3.4: Vehicle detection using RANSAC: the blue points indicate the sound delays at each time t, and the red points are points judged as being of "high likelihood" during the RANSAC fitting process. The orange line is the result of RANSAC fitting.

If the signal obtained by the first microphone enables us to broadly estimate the passing vehicle type, then the signal at the second microphone gives us information about any successively or simultaneously passing vehicles. For instance, the presence of frequency information corresponding to a high-amplitude signal at the second microphone would suggest a simultaneously passing vehicle in the opposite lane, whilst that of a lower-amplitude signal would suggest a successively passing vehicle in the same lane, or no other vehicle at all. The emphasized signal obtained from the combination of one of these frequency signatures at the second microphone with the frequency signature at the first microphone gives us information about both the passing vehicle type and any successively or simultaneously passing vehicles.

Figure 3.5 shows an overview of the emphasis synthesis process: the audio signals of the left and right channels are subdivided into multiple fixed-width windows with the time shift being performed in the frequency domain in order to process each window sequentially. We obtain the time-frequency domain representation of each individual window by performing a Fast Fourier Transform (FFT) on each of them sequentially, before using the speed v and the passing time t_0 obtained beforehand to calculate the appropriate value of Δt as seen in (3.4). Finally, one of the signals is shifted by Δt to

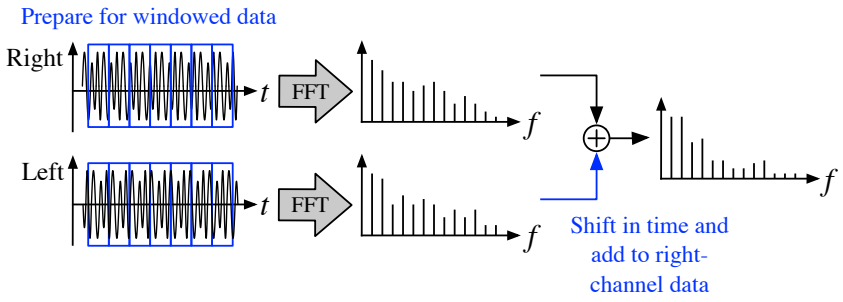


Fig. 3.5: Emphasis synthesis process: for each fixed-width window, the frequency domain representation is derived using an FFT. The FFT'd signals on left and right channels are aligned and summed together.

cancel out the time difference and the two signals are summed together. Time shifting the signal in the frequency domain amounts to shifting the phase of each of its frequency components. Let $s[n]$ be the discrete-time representation of the original signal and $S[k]$ its frequency domain representation obtained via DFT:

$$\begin{aligned} S[k] &= \text{DFT}(s[n]) \\ &= \sum_{n=0}^{N-1} s[n]\, e^{-j2\pi k \frac{n}{N}}. \end{aligned} \tag{3.5}$$

Here, DFT( ) represents the discrete Fourier transform, and N is the number of points used in the DFT operation (i.e. the window size). The DFT of the signal $s[n-m]$, which is the signal obtained by delaying the time domain representation of the signal $s[n]$ by m points, can be represented as follows:

$$\begin{aligned} \text{DFT}(s[n-m]) &= \sum_{n=0}^{N-1} s[n-m]\, e^{-j2\pi k \frac{n}{N}} \\ &= e^{-j2\pi k \frac{m}{N}} S[k]. \end{aligned} \tag{3.6}$$

From (3.6), we can see that shifting the time shifts the phase of each frequency component by $-2\pi k \frac{m}{N}$.

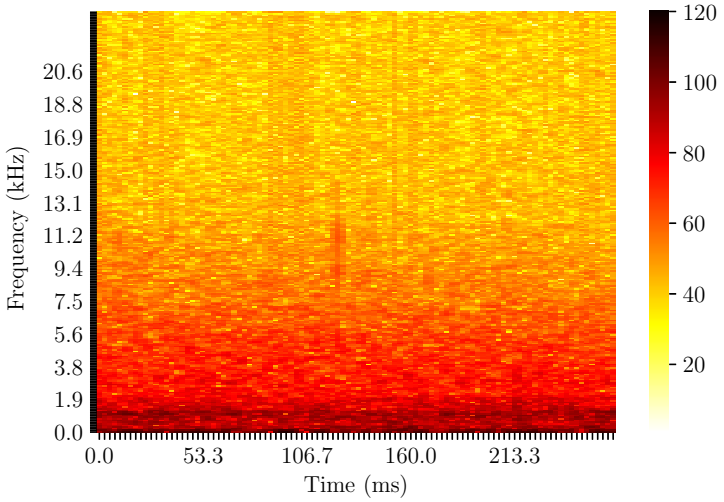
3.2.6 Vehicle Type Classification Block

The *Vehicle Type Classification* block extracts the features used for vehicle type classification from a frequency domain representation of the emphasized vehicle sound produced by the *Emphasis Synthesizer* block and determines the vehicle type using supervised learning. In our system, we use an SVM classifier due to the large number of features for each data point. The kernel used is the linear kernel, as it offers good separability for our particular dataset whilst being less complex and less prone to overfitting than other kernels.

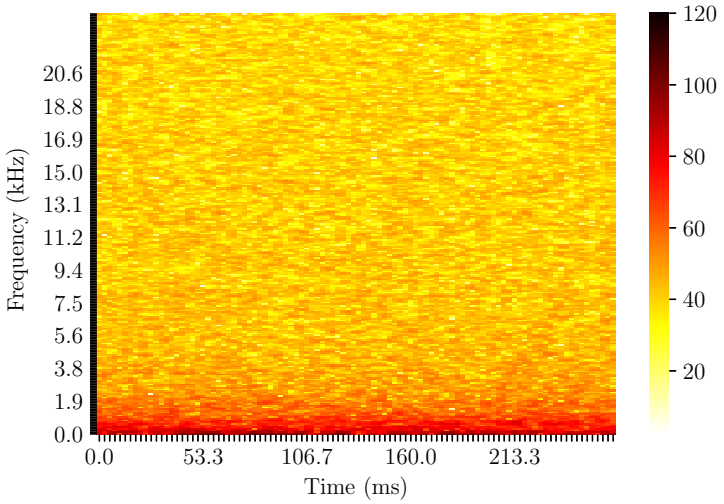
SMBAS uses only the low-frequency components of the emphasized audio signal as features. Figure 3.6 shows the frequency spectrum of two audio signals, one acquired when a vehicle was passing and one acquired when a vehicle wasn't passing. We can see that the majority of the frequency content contained in a passing vehicle's signal is located in the sub-10 kHz band.

In order to reduce the influence of environmental noise, a low-pass filter (LPF) is applied in the time domain to the individual frequency components prior to classification. Looking at the horizontal axes of Figure 3.6, we can see that whilst the frequency spectrum of the actual audio signal does not change significantly in the short period of several hundred milliseconds, there are changes in the spectrum of the signal acquired by the microphone that are due to the influence of environmental noise. Given that the time required for a vehicle to pass in front of the microphone is relatively long (on the order of a few seconds) the effect of this small change can be reduced by applying a moving average filter over a shorter time span than the vehicle passing time. Based on our preliminary experimental results, the length of the moving average is set to 320 ms in our evaluations.

Finally, to improve system accuracy and efficiency, standardization is applied to all features before classification: $(x_f[i] - \mu_f)/\sigma_f$, where $xf[i]$ is the $[i]$th entry in a feature vector, μ_f is the vector's average value, and σ_f its standard deviation.



(a) A vehicle is passing



(b) No vehicle is passing

Fig. 3.6: Sound spectrograms (dB) when a vehicle is passing, and when no vehicle is passing.

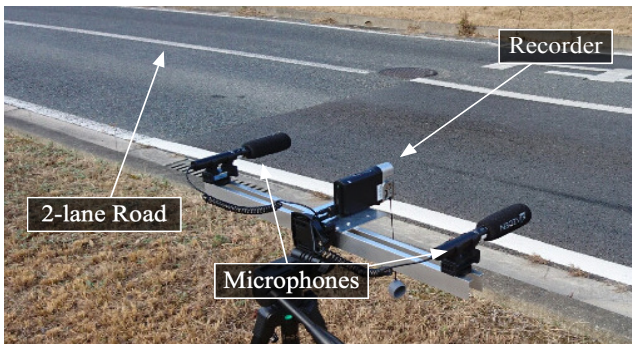


Fig. 3.7: Experimental setup: two microphones installed on the roadside record the sounds of passing vehicles.

3.3 Evaluation

3.3.1 Evaluation Environment

The experimental setup is shown in Figure 3.7. Vehicle sounds were acquired from a two-way, two-lane road and classification was performed on sounds obtained by vehicles passing on either lane. Two microphones were installed on the roadside at approximately 1 m from the ground, parallel to the road and connected to a video camera. The distance between both microphones was $D = 50$ cm, the distance between the microphones and the centre of the front lane is $L = 3$ m and the distance between the microphones and the back lane was $L = 6$ m. The video camera used was a SONY HDR-MV1 and the microphone an AZDEN SGM-990, recording for approximately 20 minutes at a sample rate of 48 kHz and bit depth of 16 bits. As in [14],

The total number of vehicles detected by the detection block was 142 vehicles (46 cars, 78 scooters/motorbikes, 18 buses), and classification was performed for 3 classes: cars, scooters/motorbikes, and buses, which we refer to as "Car", "Scooter" and "Bus" respectively.

The time taken by a vehicle to pass in front of the first microphone is defined as T_{pass}. We set the initial passing time of a vehicle in front of the microphone as $t = t_0$ and evaluate signals over the range $[t_{0,i} - T_{\text{pass}}/2;\ t_{0,i} + T_{\text{pass}}/2]$ where i corresponds to each successive passing vehicle, and $t_{0,i}$ is the initial passing time of that particular vehicle. We record each passing vehicle for $[t_{0,i} - T_{\text{pass}}/2\ ;\ t_{0,i} + T_{\text{pass}}/2]$ before splitting the acquired audio signals into a

sequence of windows which are sequentially transformed using an FFT and low-pass filtered, resulting in a spectrogram from which we extract frequency domain features (Figure 3.6). The vehicle data is randomly undersampled to obtain classes with equal amount of entries, and the features are inputted to a 10-fold cross-validated classifier.

During our evaluation we compared the classification accuracy of the following two methods:

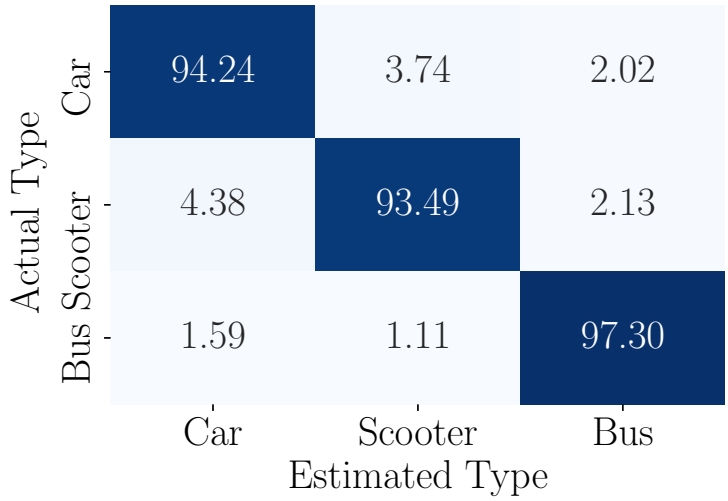
- *Stereo classification method:* our proposed method, SMBAS, illustrated in Figure 3.3. By using the information obtained during the detection process, the sound obtained by both of the microphones is combined to emphasize the vehicle sound, and the vehicle type is determined by supervised learning using features obtained from the emphasized signal.
- *Mono classification method:* this method determines the vehicle type using only one microphone. As the evaluation environment in this paper uses two microphones, in this case the vehicle type was determined using features obtained from the left microphone's audio signal only.

3.3.2 System Performance

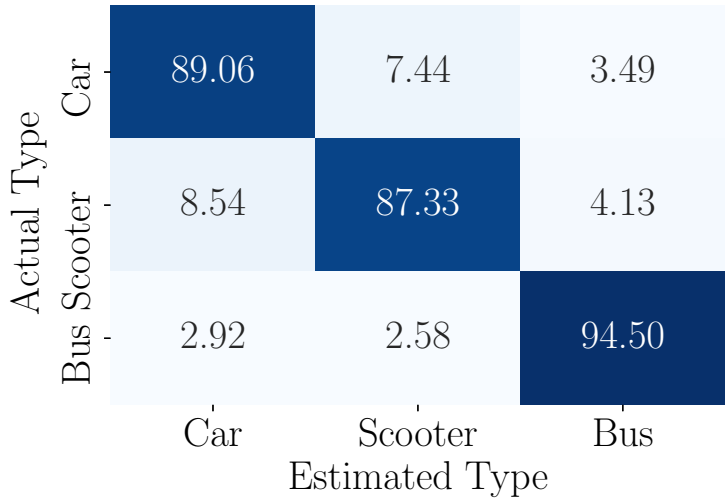
To mitigate the effects of any potential randomness due to undersampling, we run the full SMBAS system 100 times, averaging the results obtained from each run, leaving us with our final system accuracy values and confusion matrices. The FFT window length was set to 4096 points, and the features used in classification were obtained by shifting the FFT window along the captured audio signals with a 25% overlap. We set $T_{\text{pass}} = 2.0\,\text{s}$ based on the results of preliminary experiments.

Figure 3.8 shows the confusion matrices for the stereo and mono classification methods. The accuracy ratings are 95.0% and 90.3% respectively: vehicle identification accuracy is improved by 4.71 percentage points when using the stereo classification method rather than the mono classification method.

Table 3.1 shows the proportion of simultaneously and sequentially passing vehicles compared to the overall amount of passing vehicles. We define a



(a) Stereo classification method



(b) Mono classification method

Fig. 3.8: Confusion matrices for stereo and mono classification methods. Average accuracy is 95.0% and 90.3%, respectively.

vehicle as “simultaneously passing” if it passes within a previous vehicle’s T_{pass} period in the opposite direction, and “successively passing” if it passes within a previous vehicle’s T_{pass} period in the same direction.

Tab. 3.1: Number and Ratio of Successive and Simultaneous Passing Vehicles

	Car	Scooter	Bus	Total
Detected	46	78	18	142
Simultaneous	11 (23.9%)	33 (42.3%)	2 (11.1%)	46 (32.4%)
Successive	6 (13.0%)	10 (12.8%)	3 (16.7%)	19 (13.4%)
Total	17 (37.0%)	43 (55.1%)	5 (27.8%)	65 (45.8%)

The improvement in overall system accuracy can be attributed to the improved detection of simultaneously and successively passing vehicles achieved thanks to the stereo classification method. By looking at Table 3.1 and Figure 3.8 we can see that the simultaneously and successively passing vehicles make up only a relatively small proportion of the overall detected vehicles, which is why the overall system accuracy shows only a slight improvement.

3.3.3 Considerations

There are a number of important lessons to take from SMBAS which can be considered when designing our proposed C-AVDI system. First is the amount of data used: SMBAS performs detection and identification on only 142 total vehicles obtained during a single 20-minute recording session; it is important to increase this number for C-AVDI. Second, SMBAS only operates on data obtained under clear weather conditions, and has not been tested in adverse weather conditions; this also needs to be considered when designing C-AVDI. Third, the detection and identification processes are performed in two separate stages, it would be more efficient to perform both these processes in a single stage. Finally, the proportion of successively and simultaneously passing vehicles is small compared to the total amount of passing vehicles, thus focussing on the detection of such vehicles may not be the most efficient use of resources: for example, the improvement in the system efficiency achieved by using a mono input signal rather than a stereo input signal could potentially outweigh any improved in system accuracy achieved by improving simultaneous and successive vehicle detection.

3.4 Baseline Human Activity Recognition Approaches

HAR is used in a host of different scenarios, each with different constraints and objectives, and as such there is an equally large number of implementations designed to be used in these scenarios. The most common approaches can be broadly separated into two categories, remote sensor-based systems and wearable sensor-based systems:

- *Remote sensor-based systems:* this category includes sensors such as video cameras [52], radar [53], or Wi-Fi CSI channel information [54], and are more commonly used in applications where multiple subjects are being monitored simultaneously, such as in public spaces.
- *Wearable sensor-based systems:* this category includes wearable, on-board sensors placed directly on the subject, either in the form of custom hardware [55] or using sensors present in everyday objects like smartphones [56] and smartwatches [57], and is most often used to monitor the movements and activities of individual subjects.

In our research we focus on the second category. Wearable sensors used in HAR were initially limited to purpose built devices, but improvements in technology have led to the widespread adoption and use of sensor-equipped smartphones and smartwatches. This development caused a rise in both the number of people carrying sensors, and the period of time the sensors are carried for, significantly increasing the viability of individual subject-oriented HAR. Of the two devices, the smartwatch can be considered the most important as it is always placed in the same place, the wrist, a part of the human anatomy with multiple axes of movements, and whose movements vary significantly depending on the activities performed by the wearer, while the smartphone is typically limited to few degrees of movement due to usually being placed in a pocket, on a table, or in a bag.

As such, it follows that studies on individual subject-oriented HAR tend to use data collected from smartwatches or a combination of both smartwatch and smartphone data, paired with supervised learning to detect and identify a subject's activities. The authors of [57] present a system which uses

smartwatch accelerometer data in conjunction with a random forest (RF) classifier to identify activities associated with forestry and cable logging. Similarly, the authors of [63] present a system which uses data from both the smartwatch's gyroscope and accelerometer, as well as other onboard sensors, as inputs to a hybrid principal component analysis (PCA) and RF classification framework to detect and identify a set of daily activities. In [64] the authors use a selection of artificial neural networks (ANNs) with the gyroscope and accelerometer data obtained from smartwatch sensors to recognise daily activities, and the authors of [51] use the biometric data obtained by both the smartwatch and smartphone sensors in conjunction with a range of different supervised learning methods to identify different users based on how they perform certain activities.

3.4.1 Smartwatch-based HAR Baseline Approach

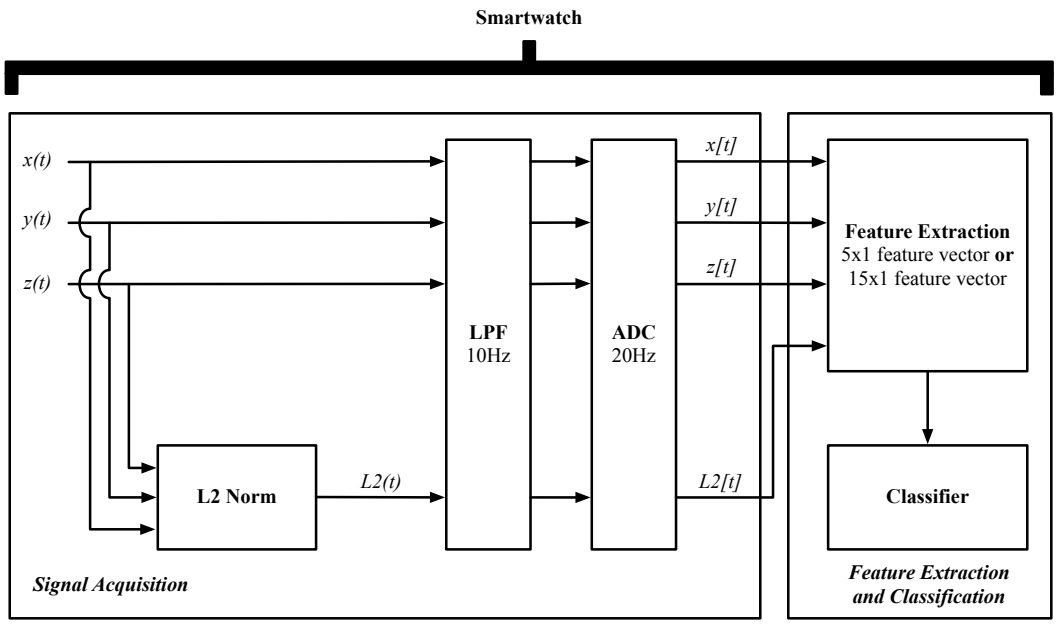


Fig. 3.9: Baseline HAR approach: we use three different approaches to extract features from the x-, y-, and z-axis sensor readings for use in classification.

At a later point in our thesis - in Chapter 6 - we propose C-HAR, a system which uses data obtained from the accelerometer and gyroscope of a smartwatch to classify a selection of activities performed by the wearer. For the sake of future comparison, in this section we present a straightforward

heuristic baseline approach. The dataset used, the WISDM dataset [51], provides three-axis sensor readings for 18 different activities, divided into three categories: "Non-Hand-Oriented", "Hand-Oriented (General)", and "Hand-Oriented (Eating)". We obtain 210 unique 3-activity combinations by drawing an activity from each category, and perform 3-class classification using a random forest (RF) classifier. We average the classification results obtained from each of the 210 individual 3-class combinations, and obtain our final accuracy values.

We use three different approaches to extract features from the x-, y-, and z-axis sensor readings for use in classification, as shown in Figure 3.9.

The extracted features comprising a 5-feature set are:

- *mean*
- *standard deviation*
- *median*
- *absolute largest value*
- *interquartile range*

We list the different approaches along with their respective highest overall sensor accuracy:

- *Individual axis:* the system extracts a set of 5 features from each axis, and uses each set in turn as inputs to a classifier. We obtain an accuracy of 97.2% using x-axis features, 95.6% using y-axis features, and 95.4% using z-axis features.
- *Combined axis:* the system extracts a set of 5 features from each axis, and introduces them as a combined 15 feature input vector to a classifier. We obtain an accuracy of 98.8%.
- *Normed axis:* the system takes the L2 norm of the three input signals and extracts a set of 5 features from the combined signal to use as inputs to a classifier. We obtain an accuracy of 96.6%.

The highest accuracy of 98.8% is obtained using the *Normed axis* approach.

We summarise the smartwatch-based approaches discussed above, including our heuristic baseline approach in Table 3.2.

Tab. 3.2: Existing Lightweight HAR Baseline Approaches

System	Focus	Sensor	Classification	Classifier	Accuracy
[52]	Posture Detection	Video Camera	Multiclass	MLP	96.0%
[53]	Indoor Activities	Radar	Multiclass	KMeans	85.0%
[54]	General Activities	WiFi CSI	Multiclass	SVM	97.0%
[57]	Domain-Specific Activities	Accelerometer	Multiclass	RF	86.7%
[63]	General Activities	Various	Multiclass	RF	98.5%
[64]	General Activities	Accelerometer	Multiclass	CNN	91.4%
Baseline (Best)	General Activities	Accelerometer	Multiclass	RF	98.8%

3.4.2 Considerations

We can draw a number of conclusions from the baseline HAR approaches described above to consider when designing our C-HAR system. The first is that all the systems present a very high accuracy score, regardless of their focus or sensing strategy. This means that even if C-HAR is able to operate at sub-Nyquist rates, it needs to achieve a comparably high accuracy to be considered a viable alternative. The second is that given the wide range of different sensors, setups and systems used in HAR, it is important to clearly define the goals and chosen approach of our proposed system, as they will be crucial in evaluating the performance of the final system. In our case, we will look to create a lightweight smartwatch sensor-based system capable of multiclass classification of everyday human activities.

3.5 Summary

In this chapter, we presented a selection of baseline approaches against which to compare our proposed compressive measurement-based systems which are to be presented in subsequent chapters. Given the relatively niche space that AVDI occupies, both within VDI and the field of ITS more generally, an in-depth look at a more traditional Nyquist rate AVDI architecture such as SMBAS is crucial when designing our C-AVDI system as it gives us significant

insight into which processes can be removed, adapted, or kept as is. In contrast, the wide range of lightweight HAR systems means that there are a large variety of applications and approaches against which our C-HAR system can be compared. This allows us to quickly and simply design and determine the viability of our proposed C-HAR system by comparing its operating process, accuracy and efficiency to existing lightweight HAR approaches.

4

Compressive Measurement Processing

4.1 Introduction

One of the main functionalities of our proposed framework is the simultaneous filtering and sub-Nyquist sampling of input signals, which requires the creation of a dedicated compressive measurement sampling and processing strategy. In this chapter, we first discuss how compressive sensing (CS) can be considered as a dimensionality reduction operation, before presenting our approach to simultaneous filtering and sampling using spectrally shaped bipolar sequences.

4.1.1 Background

The term "compressive signal processing" (CSP) was first coined by the authors of [65] and refers to the process of performing detection, classification, and filtering directly on the compressive measurements obtained during the CS process without prior reconstruction. This concept has been further explored and analysed in [25, 66, 67]; and while the goals and approaches differ, the fundamental idea of bypassing the reconstruction of x and instead directly leveraging ξ remains the same. This concept underpins the approach to sensing presented in this thesis: rather than using CS to acquire a signal at a sub-Nyquist rate and reconstruct it at a later date, features are extracted from the compressive measurements directly and used in supervised learning applications.

If the compressive measurements can be considered as a set of data points from which features can be extracted, then it follows that appropriate pre-processing of these measurements can improve the performance of the associated supervised learning systems. This pre-processing can be performed once the measurements are obtained and is typically modelled as a sequence of matrix operations on an input matrix or vector such as in [65] or [68]. While this approach can lead to precise filtering and processing, it often requires knowledge, and thus on-system storage of Θ, the reconstruction matrix used in the recovery process, which places minimum requirements on the memory and computational capabilities of the hardware on which the processing occurs.

A different approach, and the one used in our research, is to pre-process the measurements as they are acquired. The authors of [69] first introduce the concept of matching the sampling strategy to the input signal to improve reconstruction performance when using a random demodulator (RD). By matching the frequency-domain spectra of the input signal with a specially designed pseudo-random bipolar spreading sequence, the authors are able to reduce the minimum amount of samples required for perfect reconstruction when compared to using a fully randomly generated bipolar sequence. This serves as a starting point from which to design a system capable of cutting and boosting specific frequency bands present the original input signal as they are acquired, creating a simultaneous sub-Nyquist sampling and filtering architecture. Finally, we note that in this thesis, we use the term "filtering" to refer to both the attenuation and amplification of frequencies present in a signal.

4.1.2 Related Work

Sub-Nyquist signal processing using the measurements obtained during the CS process has been explored as an alternative to traditional digital signal processing in a range of existing work. The authors of [65] present a methodology in which signal processing is performed directly on the compressive measurements obtained during the CS process. The paper demonstrates that filtering, detection and classification can be performed directly on the

compressive measurements without recovering the signal beforehand. Similarly, the authors of [25] propose a multiband compressive signal processing architecture in which information acquired at sub-Nyquist rates from different frequency bands is used in a range of applications without prior signal reconstruction.

In addition, there is existing research that focuses on the creation of adaptive sampling strategies used to optimize the performance of CS-based systems. Both [69] and [70] propose a method for shaping the spectra of the pseudorandom bipolar sequences used in various CS architectures to improve signal reconstruction performance. The first proposes a Markov chain-based method inspired by run-length limited (RLL) sequences to both shape the spectrum of the bipolar sequence and limit the rate at which it switches polarity. This paper lays the groundwork for subsequent research, but does not fully explore the filtering effects of such a sequence in the context of CS-based sensing, instead focusing on optimizing signal reconstruction. The second proposes a method based on convex relaxation to create a binary sequence whose spectrum resembles that of a notch filter. While this method is able to produce sequences with well-defined spectra, the complexity associated with this method makes it unsuitable for low-power applications.

The above work serves both as a proof of concept for CSP as a lightweight alternative to traditional digital signal processing (DSP), and lays the groundwork for our simultaneous sub-Nyquist sampling and filtering classification architecture.

4.2 Compressive Sensing as a Dimensionality Reduction Operation

The computational complexity associated with the CS procedure occurs predominantly during the reconstruction process, and bypassing this procedure by performing classification based on information extracted directly from the compressive measurements ξ enables us to significantly reduce the processing requirements of our system compared to more traditional CS-based systems.

In essence, we consider the procedure $\mathbf{\Phi}$: $\mathbb{R}^N \rightarrow \mathbb{R}^M$ purely as a dimensionality reduction operation as it produces a lower dimension representation of our input signal. For us to be able to consider compressive measurements as low-dimensionality representations of an input signal, we must ensure that certain conditions are met, most importantly that for any two distinct signals x_1 and x_2, $\mathbf{\Phi}x_1 \neq \mathbf{\Phi}x_2$, and thus $\mathbf{\Theta}\alpha_1 \neq \mathbf{\Theta}\alpha_2$. This is guaranteed if $\mathbf{\Phi}$ and $\mathbf{\Theta}$ follow the criteria outlined in [19]. Furthermore, as stated in Section 2.2.1, ensuring that these two matrices follow the aforementioned criteria guarantees that the information contained in α (and thus, x) is present in the measurements ξ.

4.3 Compressive Sensing with Spectrally Shaped Bipolar Sequences

In the traditional RD architecture, the $PRS(t)$ used to demodulate the input signal switches polarity with equal probability; as a result, its frequency-domain representation resembles that of white noise. This ensures equal spreading of all K non-zero elements contained within $x(t)$ across the frequency spectrum, which is optimal if we do not have prior knowledge of the distribution of $x(t)$'s frequency information within its bandlimit.

If, however, we do have prior knowledge of the locations of interest, then [69] shows that signal reconstruction accuracy can be improved by using bipolar sequences whose frequency-domain representation matches that of the input signal. The authors argue that, given that the RD modulates the input signal with a pseudorandom sequence and low-pass filters the resulting signal, it follows that the reconstruction performance improves if more energy from the input signal is modulated to baseband. While the RD sends every spectral region to baseband with (on average) equal weighting, which is ideal if one has no prior knowledge about the frequency content of the input signal, a spectrally-tailored bipolar sequence will favour some frequency content over others in the input signal. This ensures that the modulated signal, on average, contains a large amount of relevant energy at baseband, thus improving reconstruction performance.

An example illustration of the signal matching process as described by the authors is shown in Figure 4.1. In this example, both the input signal to be acquired and the bipolar sequence used to demodulate it have their high amplitude frequency content in the same location within their bandlimits: to the left of a *frequency pivot point* at 200 Hz. As we will see later on in this chapter, however, there are situations in which it is more beneficial to match the bipolar spreading sequence's spectra to a specific band of interest in the input signal, rather than just simply matching the high amplitude frequency content locations of the two signals. The same authors demonstrate in their follow-up work [71] that the bipolar sequence can be generated using a Markov chain with each state corresponding to an output symbol of ± 1. Thus, the polarity of $PRS(t)$ at a given time is determined by the corresponding Markov chain's state transition probability and chain length. The authors also establish that the $\mathbf{\Phi}$ matrix obtained as a result of using a Markov-chain generated $PRS(t)$ satisfies the required RIP conditions outlined previously with very high probability.

These results form the starting point from which we design the simultaneous sampling and filtering section of our proposed compressive measurement processing approach presented in this chapter. We expand upon the single Markov chain sequence generation method proposed in [71] by designing a dual Markov chain sequence generation method, capable of creating bipolar sequences with more complex spectra, and prove that sequences created in this manner can be used to amplify and attenuate the frequency contents of input signals in addition to simply improving signal recovery performance.

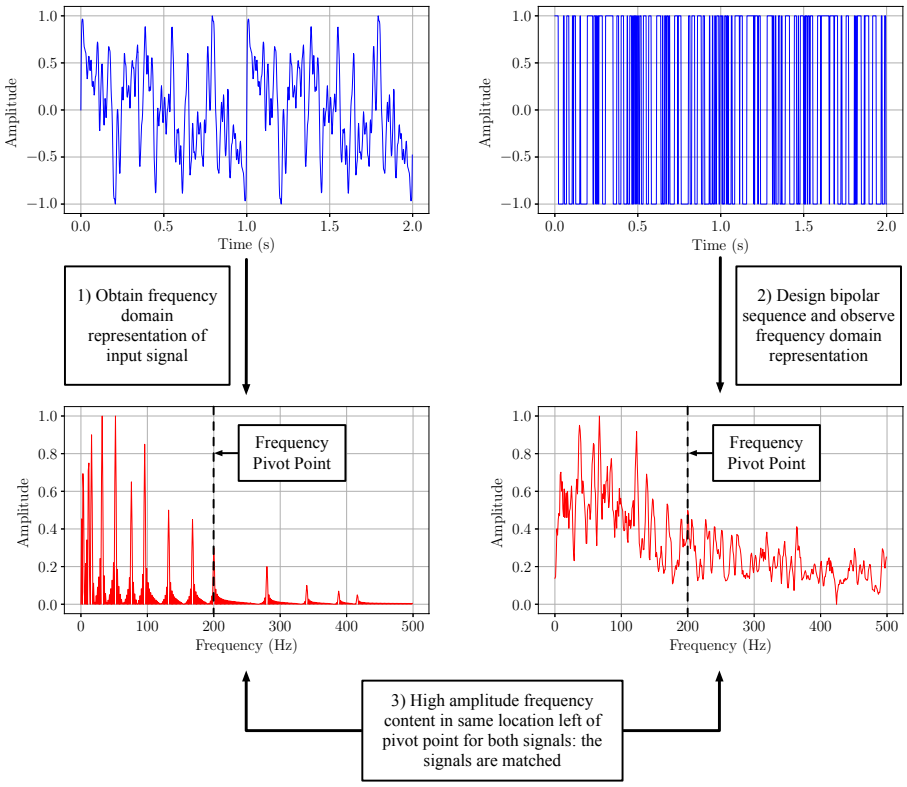


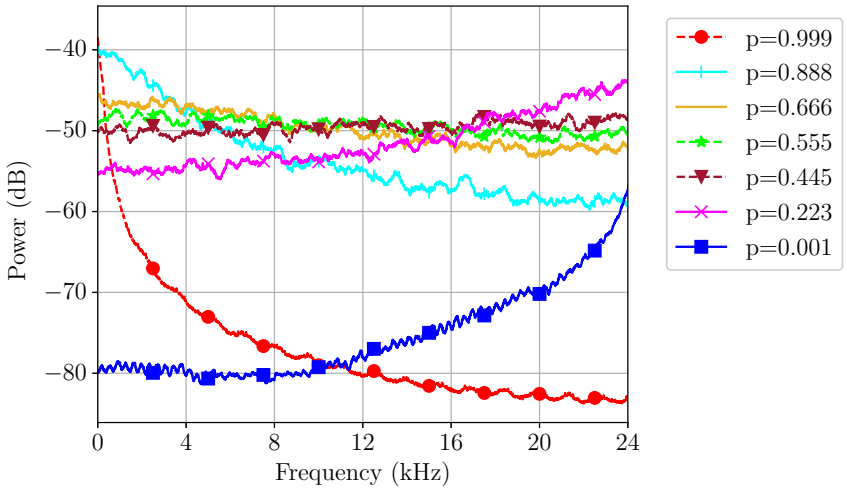
Fig. 4.1: Signal matching process: both $x(t)$ and $PRS(t)$ have their high amplitude frequency content to the left of a *frequency pivot point* at 200 Hz.

4.3.1 Markov Chain-based Generation

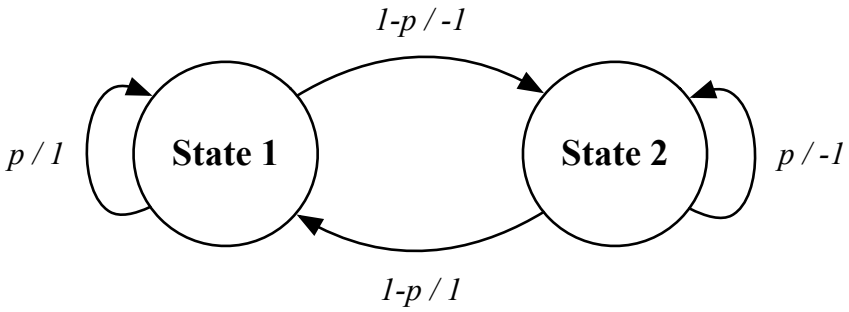
Figure 4.2 and Figure 4.3 show a selection of spectrally shaped bipolar sequence spectra and the diagrams of the corresponding Markov chains used to create them. The transition probability matrices corresponding to the 2-state and 4-state Markov chains are defined as $\mathbf{P_1}$ and $\mathbf{P_2}$ respectively:

$$\mathbf{P_1} = \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix} \mathbf{P_2} = \begin{pmatrix} 0 & p & 1-p & 0 \\ 0 & 0 & 1 & 0 \\ 1-p & 0 & 0 & p \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{4.1}$$

We sweep the value of the transition probability p over the range $0 < p < 1$. The higher the value of p, the more likely the 2-state chain is to stay in

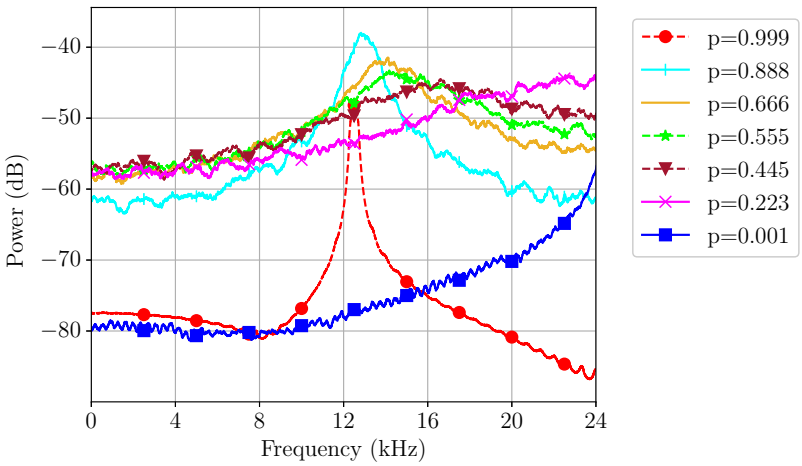


(a) Spectra of 2-state Markov chain-generated bipolar sequences

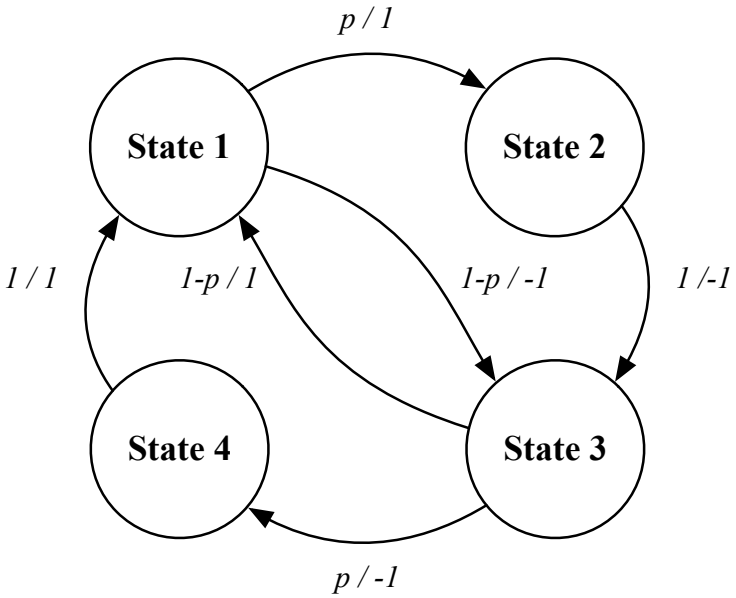


(b) State transition diagram of 2-state Markov chain-generated bipolar sequences

Fig. 4.2: Spectra and corresponding state transition diagram of 2-state Markov chain-generated bipolar sequences. Each spectrum corresponds to a different transition probability p.



(a) Spectra of 4-state Markov chain-generated bipolar sequences



(b) State transition diagram of 4-state Markov chain-generated bipolar sequences

Fig. 4.3: Spectra and corresponding state transition diagram of 4-state Markov chain-generated bipolar sequences. Each spectrum corresponds to a different transition probability p.

its current state, and the more likely the 4-state chain is to transition to a state with the same output. In the case of a 2-state chain, this results in more of the generated signal's energy being located towards the lower end of its bandlimit; and in the case of a 4-state chain this results in the generated signal's energy tending towards a narrow peak in the middle of its bandlimit. Conversely, a lower value of p means that both chains are more likely to transition to a state with a different output, leading to the energy of both of the generated signals being located towards the higher end of their respective bandlimits. It is important to note that this relationship is due to the way $\mathbf{P_1}$ and $\mathbf{P_2}$ were designed: permuting the rows and columns of these matrices would change our state diagram and reverse the relationship between frequency distribution and transition probability.

4.3.2 Reconstruction Bounds

As stated previously, in our proposed framework we do not reconstruct the original input signal at any point during its actual operation. When designing the system, however, it is important to quantify and visualize the effects of matching $PRS(t)$ and $x(t)$, which can be done most effectively by reconstructing test signals acquired using spreading sequences generated with a range of different bipolar sequence generation parameters and examining the results.

This is done by performing CS using the RD on a set of four test signals whose spectra are shown in Figure 4.4, varying the Markov chain's p-value over each run, and repeating the process for both a 2-state and a 4-state chain. The test signals each have the same bandlimit but different frequency compositions: a low-frequency x_{LF} signal, mid-frequency x_{MF} signal, high-frequency x_{HF} signal, and broadband x_{BB} signal. We denote the corresponding reconstructed versions of these signals as $\hat{x}_{LF}$, $\hat{x}_{MF}$, $\hat{x}_{HF}$ and $\hat{x}_{BB}$ (reconstruction is performed via basis pursuit using the SPGL1 toolbox available from [72]).

The test signal reconstruction parameters are shown in Table 4.1. The optimal value of M is determined by first establishing a theoretical lower bound value using (2.9), from which the optimal value of M is obtained experimentally by repeatedly performing the CS process on our input signals

while incrementally increasing M and noting the resulting relative error (RE). Repeating this process enables us to find the values of M after which the RE levels off for each of the four test signals. We use the largest of the four values, $M = 12000$.

Figure 4.5 plots the resulting RE of the original and reconstructed signals, and the values of p for the optimal reconstruction of each signal are summarized in Table 4.2. These results confirm that for a constant value of M, matching $PRS(t)$ and $x(t)$ by changing the value of p improves reconstruction performance.

4.3.3 Single Chain

In addition to improving reconstruction accuracy, a tailored $PRS(t)$ can amplify or attenuate specific frequency content in a given input signal. Thus, we can equate the mixing of $x(t)$ and $PRS(t)$ to a simultaneous demodulating and filtering operation in which the spectrum of the bipolar sequence is likened to the frequency response of a filter. This property is a key feature of our proposed system, and again is most effectively quantified and visualized during the system design process by reconstructing the test signals shown in Figure 4.4.

Tab. 4.1: Test Signal Reconstruction Parameters

Nyquist Rate W	48 kHz
Switching Rate C	48 kHz
ADC Rate R	12 kHz
Bit Depth B	12 bits
Signal Length (Time) T_s	$1s$
Signal Length (Samples) N	48000
Signal Sparsity K	40
Compressive Measurements M	12000

We visualize the filtering effects of a spectrally tailored $PRS(t)$ on an input signal by plotting the spectra of test signals $\hat{x}_{LF}$, $\hat{x}_{MF}$, $\hat{x}_{HF}$ and $\hat{x}_{BB}$ recon-

Tab. 4.2: Single Chain Parameters for Optimal Reconstruction

Chain Length	Signal	p	RE [$\times 10^{-5}$]
2-state	$\hat{x}_{LF}$	0.999	6.67
	$\hat{x}_{MF}$	0.445	10.0
	$\hat{x}_{HF}$	0.001	6.80
	$\hat{x}_{BB}$	0.555	12.7
4-state	$\hat{x}_{LF}$	0.555	10.7
	$\hat{x}_{MF}$	0.999	8.00
	$\hat{x}_{HF}$	0.001	6.80
	$\hat{x}_{BB}$	0.445	17.2

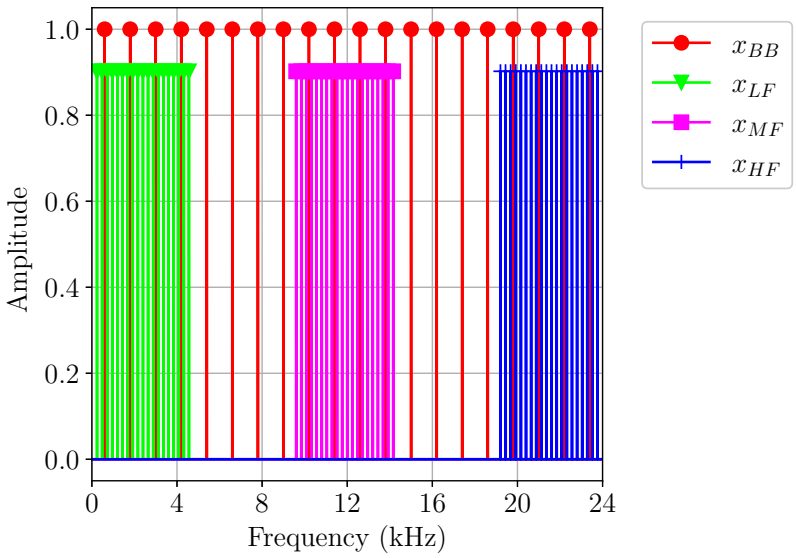
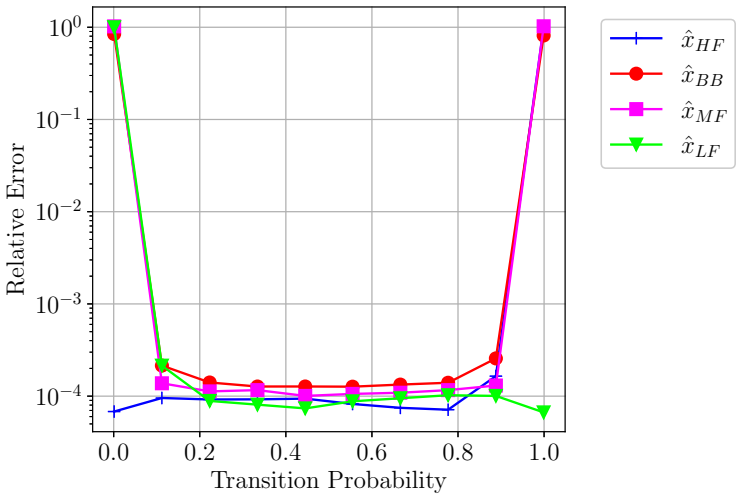
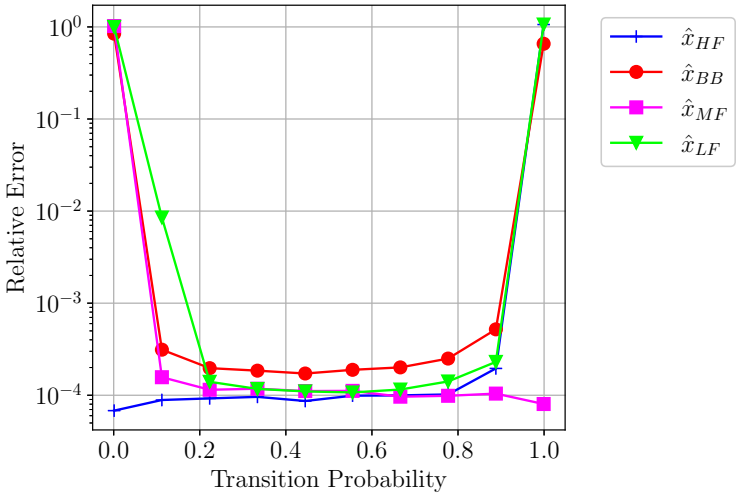


Fig. 4.4: Spectra of test signals x_{LF}, x_{MF}, x_{HF}, and x_{BB}: each signal has the same bandlimit but a different frequency composition (low-frequency, mid-frequency, high-frequency, and broadband).

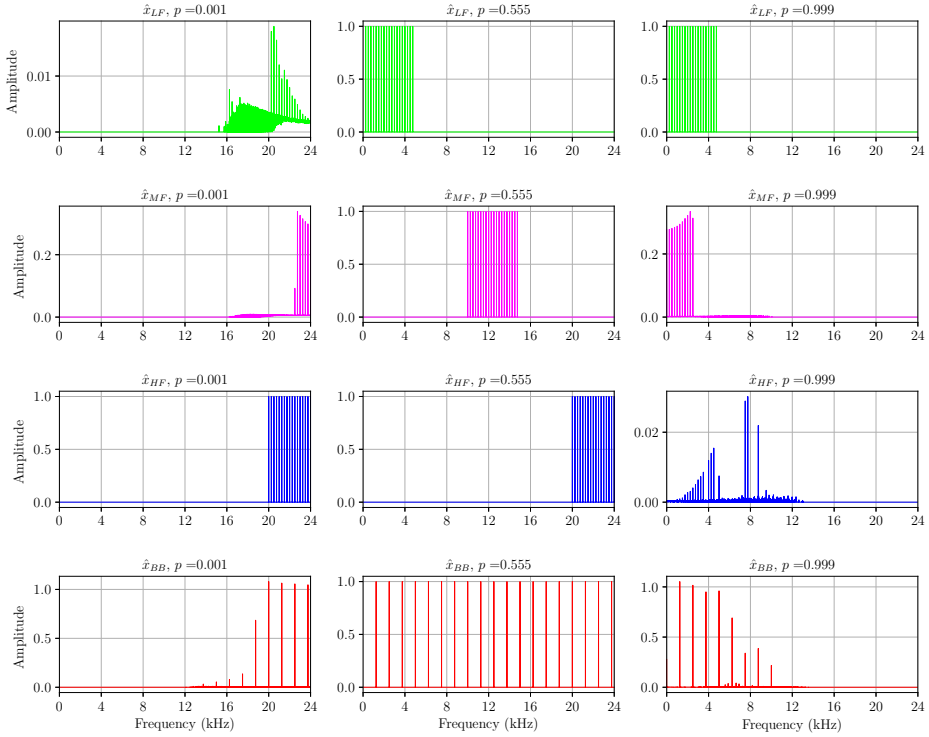


(a) Relative error as a function of transition probability for a 2-state Markov chain-generated $PRS(t)$

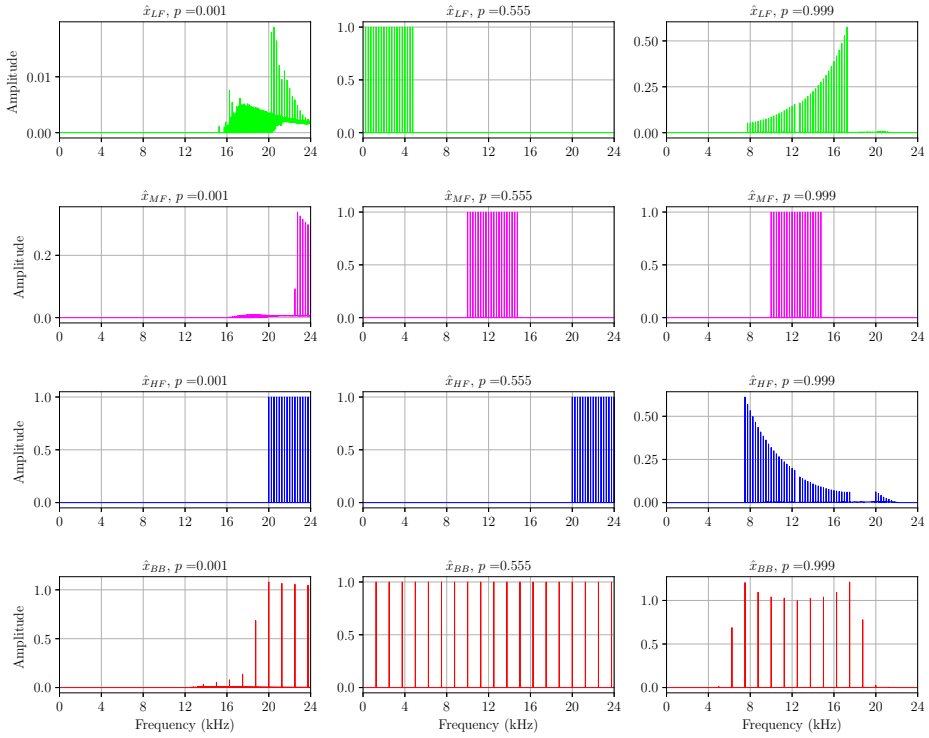


(b) Relative error as a function of transition probability for a 4-state Markov chain-generated $PRS(t)$

Fig. 4.5: Relative error as a function of transition probability for a 2-state and a 4-state Markov chain-generated $PRS(t)$.



(a) $\hat{x}_{LF}$, $\hat{x}_{MF}$, $\hat{x}_{HF}$, $\hat{x}_{BB}$ using 2-state Markov chain-generated $PRS(t)$



(b) $\hat{x}_{LF}$, $\hat{x}_{MF}$, $\hat{x}_{HF}$, $\hat{x}_{BB}$ using 4-state Markov chain-generated $PRS(t)$

Fig. 4.6: Reconstructed test signals $\hat{x}_{LF}$, $\hat{x}_{MF}$, $\hat{x}_{HF}$, and $\hat{x}_{BB}$ using a) a 2-state, and b) a 4-state Markov chain-generated $PRS(t)$ with transition probability p. Each row represents one of the four test signals presented in Figure 4.4, and each column represents a different value of p.

structed using different $PRS(t)$ with Markov chain transition probability values $p = 0.01$, $p = 0.555$, and $p = 0.999$.

Figure 4.6 shows the test signal spectra reconstructed using a 2-state chain and a 4-state chain respectively. We can see that the filtering effect a bipolar sequence has on a test signal depends on the length and p-value of the Markov chain used to generate it. This is best illustrated by x_{BB}: for a 2-state chain-generated $PRS(t)$, a value of $p = 0.001$ suppresses the low frequency content of $\hat{x}_{BB}$, and a value of $p = 0.999$ suppresses the high frequency content; for a 4-state chain-generated $PRS(t)$, a value $p = 0.001$ again suppresses the low frequency content of $\hat{x}_{BB}$, and a value of $p = 0.999$ suppresses the high and low frequency content. The filtering effect is consistent with the shape of the $PRS(t)$ spectra shown in Figure 4.2 and Figure 4.3.

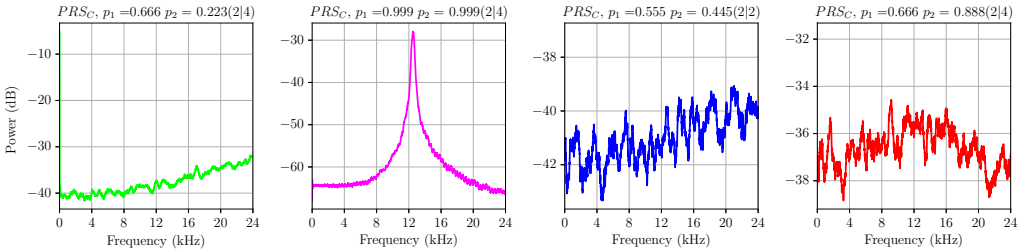
4.3.4 Dual Chain

Tab. 4.3: Dual Chain Parameters for Optimal Reconstruction

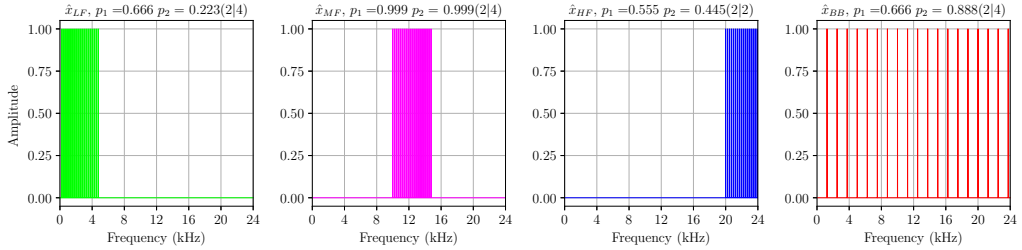
Chain 1st	Chain 2nd	Signal	p_1	p_2	RE $[\times 10^{-5}]$
2-state	2-state	$\hat{x}_{LF}$	0.666	0.001	6.64
		$\hat{x}_{MF}$	0.888	0.555	9.20
		$\hat{x}_{HF}$	0.555	0.445	6.64
		$\hat{x}_{BB}$	0.666	0.445	12.2
2-state	4-state	$\hat{x}_{LF}$	0.666	0.233	6.52
		$\hat{x}_{MF}$	0.999	0.999	7.98
		$\hat{x}_{HF}$	0.999	0.001	6.81
		$\hat{x}_{BB}$	0.666	0.888	11.2
4-state	2-state	$\hat{x}_{LF}$	0.223	0.666	6.52
		$\hat{x}_{MF}$	0.999	0.999	7.98
		$\hat{x}_{HF}$	0.001	0.999	6.81
		$\hat{x}_{BB}$	0.888	0.666	11.2
4-state	4-state	$\hat{x}_{LF}$	0.999	0.999	6.68
		$\hat{x}_{MF}$	0.999	0.001	8.12
		$\hat{x}_{HF}$	0.555	0.445	7.00
		$\hat{x}_{BB}$	0.888	0.666	11.3

In order to create $PRS(t)$ with more complex spectra, we design combined dual Markov chain-generated sequences by mixing two different single chain sequences, which we define as $PRS_C(t)$, $PRS_1(t)$, and $PRS_2(t)$ respectively, where $PRS_C(t) = PRS_1(t)PRS_2(t)$.

To fully assess the effects of $PRS_C(t)$ on a given input signal, we perform three reconstruction tests using the x_{LF}, x_{MF}, x_{HF} and x_{BB} signals shown in

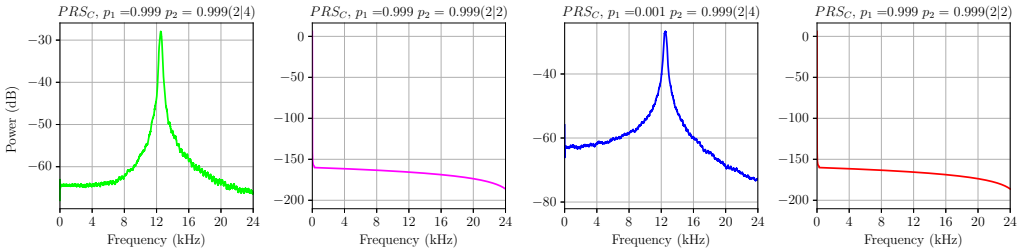


(a) First reconstruction test: PRS_C spectra

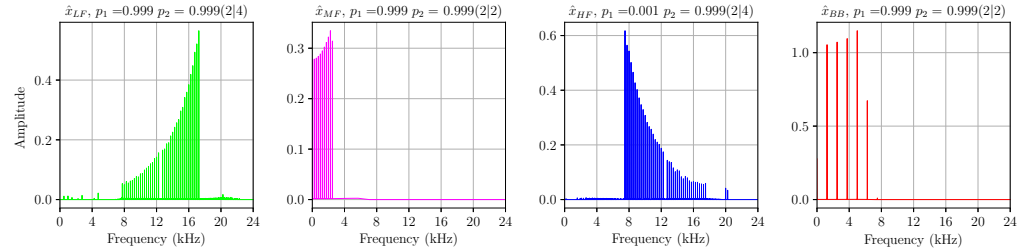


(b) First reconstruction test: $\hat{x}_{LF}$, $\hat{x}_{MF}$, $\hat{x}_{HF}$, $\hat{x}_{BB}$ spectra

Fig. 4.7: First reconstruction test: spectra of PRS_C and reconstructed test signals $\hat{x}_{LF}$, $\hat{x}_{MF}$, $\hat{x}_{HF}$, and $\hat{x}_{BB}$, with the corresponding p_1, p_2, and respective chain lengths indicated in parentheses above each subplot.

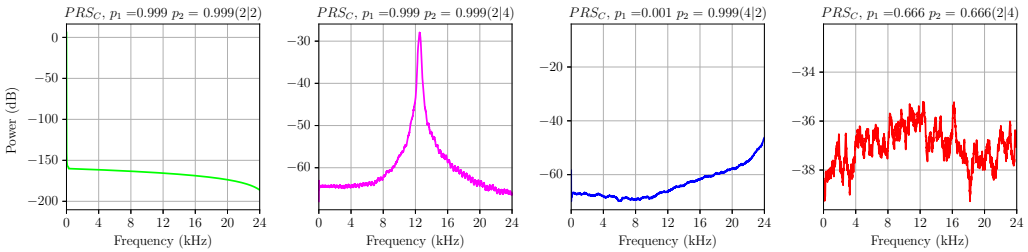


(a) Second reconstruction test: PRS_C spectra

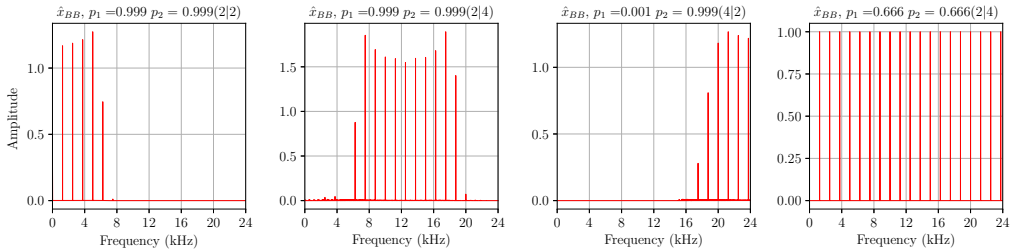


(b) Second reconstruction test: $\hat{x}_{LF}$, $\hat{x}_{MF}$, $\hat{x}_{HF}$, $\hat{x}_{BB}$ spectra

Fig. 4.8: Second reconstruction test: spectra of PRS_C and reconstructed test signals $\hat{x}_{LF}$, $\hat{x}_{MF}$, $\hat{x}_{HF}$, and $\hat{x}_{BB}$, with the corresponding p_1, p_2, and respective chain lengths indicated in parentheses above each subplot.



(a) Third reconstruction test: PRS_C spectra



(b) Third reconstruction test: $\hat{x}_{BB}$ spectra

Fig. 4.9: Third reconstruction test: spectra of PRS_C and reconstructed test signal $\hat{x}_{BB}$, with the corresponding p_1, p_2, and respective chain lengths indicated in parentheses above each subplot.

Figure 4.4, while varying the p-values of $PRS_1(t)$, $PRS_2(t)$ (p_1 and p_2 respectively), and chain lengths. The chain lengths are indicated in parentheses above each subplot, with the first number denoting the first chain's length, and the second number denoting the second chain's length. The purpose of these three reconstruction tests is to visualize different properties of our proposed dual chain-generated $PRS_C(t)$ approach:

- *First reconstruction test:* The purpose of the first reconstruction test is to gauge the improvements in signal reconstruction when using a dual chain-generated bipolar sequence. This is achieved by determining the optimal $PRS_C(t)$ that minimizes the reconstruction RE for each of the four test signals and comparing their REs to those of their single chain-generated counterparts shown in Table 4.2. The spectra of the optimal dual chain-generated sequences and of the reconstructed test signals are shown in Figure 4.7, with the optimal dual chain reconstruction parameters summarized in Table 4.3. These results show that using dual chain-generated sequences improves reconstruction performance compared to single chain-generated sequences, even if only marginally.

- *Second reconstruction test:* The purpose of the second reconstruction test is to visualize the effects of any mismatches between the input signal and dual chain-generated sequence. This is achieved by reconstructing the four test signals using $PRS_C(t)$ designed to maximize reconstruction RE and comparing the spectra of the original and reconstructed signals. The spectra of the reconstructed signals and of their respective sequences are shown in Figure 4.8. These results show the impact of a mismatched $PRS_C(t)$ on signal reconstruction, underlining the importance of using an appropriately designed $PRS_C(t)$.

- *Third reconstruction test:* The purpose of the third reconstruction test is to visualize the filtering properties of the $PRS_C(t)$. We reconstruct the broadband signal, x_{BB}, using four $PRS_C(t)$ sequences whose spectra can be likened to a low-pass, band-pass, and high-pass filter for the first three signals, and with the spectrum of the final signal resembling white noise. The spectra of the bipolar sequences and of the reconstructed test signals are shown in Figure 4.9. These results show that in addition to having a direct impact on reconstruction performance, a tailored $PRS_C(t)$ can attenuate or amplify the frequency content of the input signal.

4.4 Results and Discussion

The results obtained in this chapter confirm that, with prior knowledge of the spectral locations of interest in the input signal, using tailored bipolar sequences during the demodulation process achieves better reconstruction performance than using a fully random sequence. Conversely, using a mismatched sequence can worsen reconstruction accuracy, and that in cases without prior knowledge of the spectral locations of interest, it is better to demodulate using a fully random signal. Moreover, we showed that the spectral makeup of the PRS has a filtering effect on the input signal, allowing us to attenuate and amplify locations of interest, improving subsequent classification accuracy.

The PRS sequences were generated using a combination of single and dual Markov chains, whose frequency-domain representations are determined

by the chains' lengths and transition probabilities. This ensured that the resulting $\mathbf{\Phi}$ and $\mathbf{\Theta}$ matrices adhered to the strict criteria outlined in CS theory, thereby allowing the $\xi[m]$ measurements to be considered as a reduced-dimensionality representation of the input signal.

The Markov chain-based bipolar spreading sequence generation methodology described in this chapter underpins the two compressive-measurement based classification systems presented in the following Chapter 5 and Chapter 6. In both applications, the simultaneous sub-Nyquist filtering and sampling of input signals is a key factor in making both C-AVDI and C-HAR viable lightweight alternatives to existing systems.

4.5 Summary

In this chapter, we presented a simultaneous sub-Nyquist sampling and filtering approach based on the RD architecture. We showed that the frequency spectrum of the demodulating signal $PRS(t)$ has a direct effect on the frequency spectrum of the reconstructed signal $\hat{x}(t)$, allowing us to simultaneously acquire and filter the input signal $x(t)$. Generating the bipolar sequences using Markov chains ensures that the fundamental underlying properties of CS remain valid in our approach.

The results presented in this section play a fundamental role in the design and operation of our C-AVDI and C-HAR systems presented in Chapter 5 and Chapter 6.

5

C-AVDI: Compressive Measurement-Based Acoustic Vehicle Detection and Identification

5.1 Introduction

In this chapter, we present our compressive measurement-based acoustic vehicle detection and identification system (C-AVDI), the first of our two proposed compressive measurement-based lightweight classification systems. We begin by describing the vehicle sound dataset used in our approach, before giving an overview of C-AVDI's operation, followed by presenting the system's evaluation results.

5.1.1 Background

In recent years, we have seen a rise in the development and adoption of intelligent transportation systems (ITS) technology. Key applications such as traffic flow control, navigation systems, and road safety management are growing ever more sophisticated and gaining increasingly widespread use. These improvements, however, come at a cost: the increasing amount of data created, processed, and used requires expensive, power-intensive hardware to be stored, accessed, and leveraged effectively. Particularly important in ITS is the vehicle detection and identification (VDI) process, a key functionality

underpinning a significant number of applications such as traffic flow and congestion management, electronic toll collection (ETC), and transportation infrastructure monitoring. Lowering the cost and computational requirements associated with the VDI process will have a direct effect on the overall cost and complexity of many ITS applications.

Various lightweight VDI techniques have been proposed, with acoustic vehicle detection and identification (AVDI) in particular being the subject of wide-ranging research. The low price and simple installation process of AVDI systems make them an attractive alternative to other more expensive and difficult-to-install VDI systems, such as video camera-, radar-, or induction loop coil-based systems.

While the installation costs associated with AVDI systems are generally low, the subsequent analysis and leveraging of data is often relatively costly in terms of computational requirements, making the use of such systems in power-critical applications difficult. In current AVDI systems, the computational costs are incurred due to processes occurring in either the acquisition and preprocessing stage as in [40] and [43] (successive discrete Fourier transforms (DFTs) or discrete wavelet transforms (DWTs)), or the classification stage as in [73] and in [74] (use of multilayer perceptron (MLP) and artificial neural network (ANN) respectively).

If we are to fully leverage the advantages of AVDI, it is necessary to find a way to reduce the computational burden associated with the data processing and analysis. To this end, we propose C-AVDI, our compressive measurement-based acoustic vehicle detection and identification system.

5.1.2 Related Work

In Section 3.2.1, we explored some existing VDI approaches, and briefly explained the operation of some vehicle detection, and speed and direction estimation systems. In this section, we focus specifically on existing research that uses supervised learning techniques to detect and identify vehicles using features extracted from their acoustic signatures.

The frequency-domain features extracted from vehicle audio signals are used with a support vector machine (SVM) classifier to identify and classify vehicles and their associated parameters in [44], and used to detect vehicles for the purpose of collision avoidance in non-line-of-sight situations in [75]. Mel-frequency cepstral coefficients (MFCCs) are used in conjunction with machine learning (ML) or deep learning (DL) as features in a number of existing AVDI systems: in [41] they are used with a modified MLP, in [74] they are extracted from a specific high energy audio region and used with an ANN and k-nearest neighbours (KNN) classifier, and in [76] they are used in a feature set containing the pitch class profile (PCP) and short-term energy (STE) of vehicle audio signals in a hybrid convolutional neural network (CNN) containing a long short-term memory (LSTM) layer. In [73], Göksu presents a system capable of analysing the acoustic signatures of vehicles independently of any changes in engine sound. Using wavelet packet decomposition (WPD) and an MLP classifier, the system obtains engine speed-independent features from the acoustic signals of passing vehicles. On the other hand, the system in [77] identifies different vehicles based on the sound of their engines using modulated per-channel energy normalization (Mod-PCEN) features in tandem with a Siamese neural network (SNN).

All of these aforementioned existing AVDI systems share a common, or very similar, goal and basic approach, but differ in their implementation, applications, and performance. While these systems generally all present good performance metrics, the use of computationally intensive input signal processing or a complex supervised learning method offsets the improvements in system cost, complexity, efficiency and flexibility.

The authors of [47] propose an ultra-low power vehicle detector (ULP-VD) capable of detecting passing vehicles with minimal computational cost using logistic regression (LR). This system, however, is only able to detect the presence of passing vehicles and requires an additional stage to identify them.

We ourselves have previously presented two AVDI approaches. The first system is SMBAS [40], our baseline approach covered in Section 3.2. Using features extracted from an emphasized mono signal produced by combining the two time-shifted signals of a stereo microphone pair, it obtains an

Tab. 5.1: Summary of Related AVDI Work

System	Can Detect	Can Identify	Run on MCU	Input Signal Processing	Detection /Identification
[39]	✓	✓		CS	RF
[40]	✓	✓		STFT	SVM
[73]	✓	✓		WPD	MLP
[74]	✓	✓		MFCC	KNN /ANN
[44]	✓	✓		DFT	SVM
[75]	✓			STFT	SVM
[41]	✓	✓		MFCC	MLP
[76]	✓	✓		MFCC /PCP/STE	CNN /LSTM
[77]	✓	✓		Mod-PCEN	SNN
[47]	✓		✓	DWT	LR

accuracy of 95.0% with classification being performed using an SVM classifier. The second is our initial attempt at creating a lightweight compressive measurement-based AVDI system, presented in [39]. By combining a more traditional sub-Nyquist sampling architecture with a tailored analogue front-end filtering section, the system was able to identify and detect vehicles with an accuracy of 82.6%, and a back-end sampling rate of 3 kHz using a random forest (RF) classifier. The front-end filtering section is an integral part of the system; however, its implementation as a separate stage required the use of additional components, increasing deployment and implementation costs.

The main characteristics of AVDI systems described above are summarized in Table 5.1. To the best of our knowledge, there is no current AVDI system capable of performing both detection and identification on an MCU.

5.1.3 Goals and Objectives

Our objective is to create a lightweight alternative to existing AVDI sensing systems that can implemented on a microcontroller (MCU) and whose performance is comparable to that of currently available systems. This is achieved first and foremost by reducing the sample rate at which the vehicle sounds are acquired. Indeed, the biggest limitation of MCUs is the amount of available volatile and nonvolatile memory, and while this is not a problem when working with short signals such as in [11], it makes it impossible to use MCUs in AVDI applications where the signal length is typically on the order of

a few seconds. By reducing the number of samples, we lower the memory requirements required for both the implementation of the classifier (nonvolatile memory), and the feature extraction process (volatile memory).

We achieve this by expanding on the initial proof of concept system presented in [39] in a number of ways. We begin by removing the front-end filtering stage and instead filter the input signal directly during the acquisition process using a spectrally tailored bipolar pseudo-random sequence, following the approach outlined in Chapter 4. This reduces the number of components in the system and allows the filtering parameters to be adjusted by modifying the spectrum of the sequence rather than the hardware. We then test the system under adverse weather conditions, and observe their effects on the detection and identification performance.

Our main contributions can be summarised as follows:

- We propose a C-AVDI architecture capable of operating at sub-Nyquist rates. Leveraging the inherent structure of vehicle sound signals enables us to reduce the number of samples required to detect and identify passing vehicles by a factor of 16 while maintaining an accuracy comparable to those of existing systems.
- We simultaneously sample and filter incoming vehicle signals using spectrally shaped bipolar sequences generated from a pair of Markov chains. Filtering signals in this manner improves the classification accuracy of our system.
- Demonstrate the viability of C-AVDI as a lightweight sensing architecture by benchmarking our proposed system against our previous work.

In this chapter, we focus on creating the C-AVDI architecture by simulating and optimising the various parameters involved in the system design process in software. The MCU implementation of the feature extraction and classification processes of our proposed system is described in-depth in Chapter 7.

5.2 Experimental Data

The audio data used in the design and testing of the system is collected from vehicles traversing a university campus.

The data acquisition setup in clear, rainy, and windy conditions is shown in Figure 5.1. A pair of Azden SGM-990 microphones are installed parallel to a one-lane two-way road and connected to a Sony HDR-MV1 video camera. The microphones are 1m from the ground, the intra-microphone distance is 50 cm, the distance between the microphones and the centre of the front lane is 3 m, and the distance between the microphones and the centre of the back lane is 6 m. The microphones' pickup pattern is set to *cardioid*, and they record the sound of passing vehicles for a 30-minute duration at a sample rate of 48 kHz and a bit depth of 16 bits. The signals obtained from the two microphones are averaged to create a mono signal used in subsequent analysis.

We obtain 14 different datasets by recording vehicle sounds at the same location with the same setup on 14 different days; at different times of day; on different days of the week; and under clear, windy, and rainy weather conditions. The average vehicle signal plots using data obtained under all weather conditions are shown in Figure 5.2, and the individual average vehicle signal plots for each weather condition are shown in Figure 5.3.

The three classes considered for classification are cars, scooters/motorbikes, and no vehicles. We refer to these as "Car", "Scooter", and "NoVeh", respectively. The time at which a given vehicle passes in front of the middle of the microphone pair is defined as t_p, and the time window for the vehicle as $T_r = \left[t_p - \frac{T_s}{2}\ ;\ t_p + \frac{T_s}{2}\right]$, where $T_s = 2s$. As we are not seeking to detect successively or simultaneously passing vehicles in this evaluation, we retain only the vehicle signals whose T_r does not overlap with those of the preceding or following signals. The total number of vehicle sounds obtained for each class on each day can be seen in Table 5.2.

Tab. 5.2: Weather Conditions and Number of Vehicles for Each Day

Day	Weather	Car	Scooter	NoVeh	Total
1	Clear	60	8	104	172
2	Clear	21	10	45	76
3	Wind	21	40	64	125
4	Clear	40	57	115	212
5	Clear	52	50	112	214
6	Rain	43	26	77	146
7	Rain	77	30	152	259
8	Clear	177	95	305	577
9	Wind	174	103	267	544
10	Clear	143	99	239	481
11	Wind	129	84	234	447
12	Clear	130	59	197	386
13	Wind	119	51	186	356
14	Rain	228	65	317	610

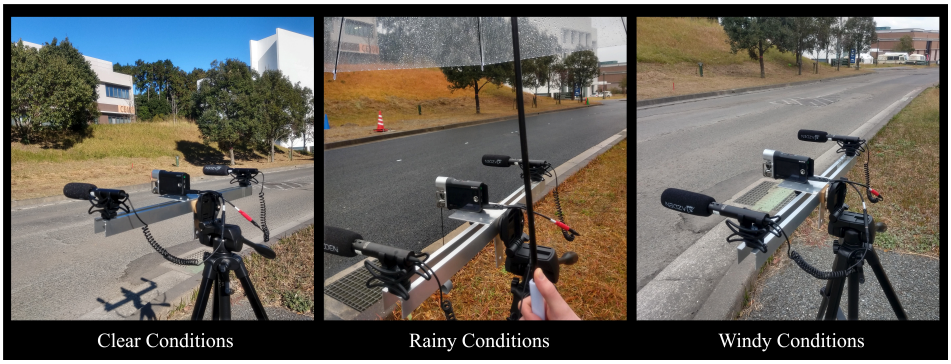


Fig. 5.1: Experimental setup: two microphones installed on the roadside record the sounds of passing vehicles in clear, rainy, and windy conditions.

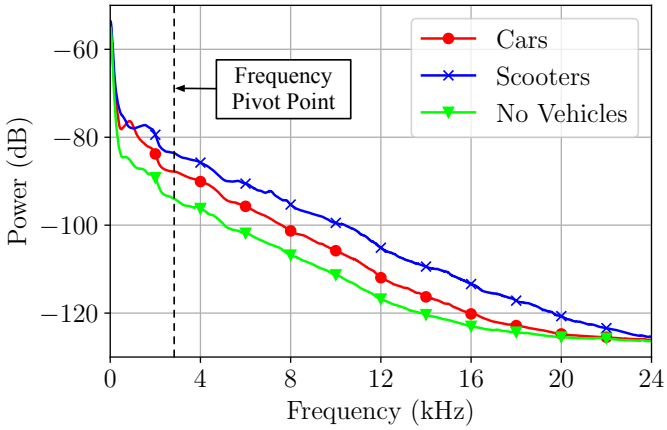
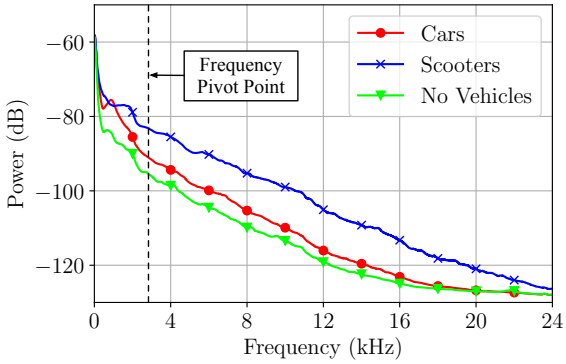
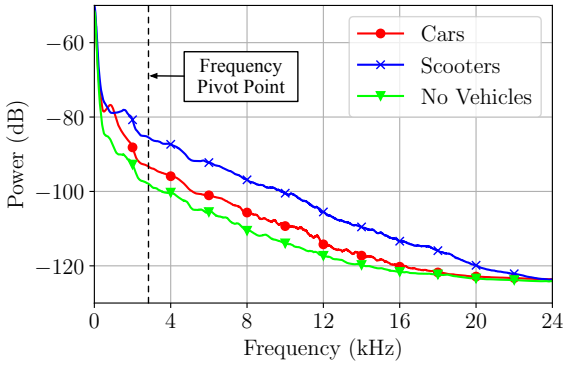


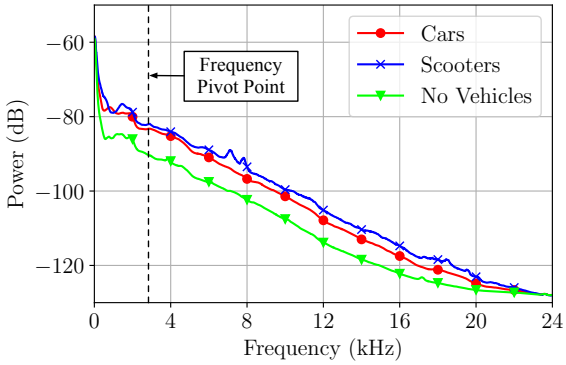
Fig. 5.2: Average audio signals for three vehicle classes.



(a) Average signals recorded in clear conditions



(b) Average signals recorded in windy conditions



(c) Average signals recorded in rainy conditions

Fig. 5.3: Average audio signals for three vehicle classes recorded in clear, windy, and rainy conditions.

5.3 System Overview

Our proposed system aims to obtain information from the audio signals of passing vehicles while sampling at sub-Nyquist rates. Features are extracted

from the samples obtained in this manner and are used to detect and identify the vehicles. Some details relating to the operation of specific sections and blocks of the system have been covered in more detail in previous chapters. We will indicate in the text when this is the case, and refer to the relevant chapter or section.

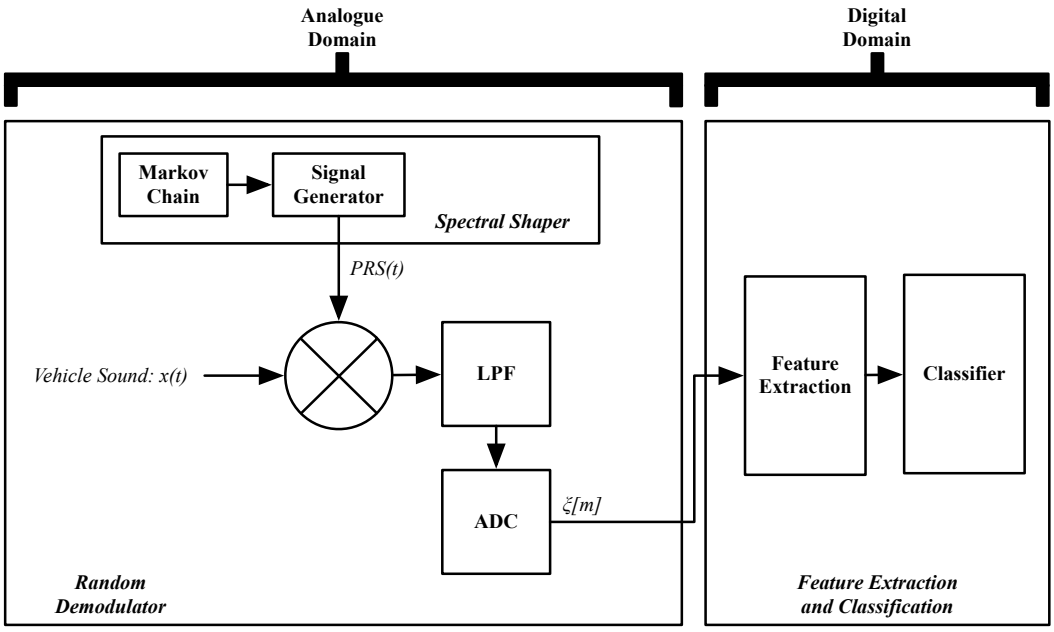


Fig. 5.4: C-AVDI system overview: an input signal $x(t)$ is simultaneously sampled and filtered during the signal acquisition process using a spectrally shaped bipolar pseudorandom sequence $PRS(t)$ generated by the *Spectral Shaper*.

An overview of our proposed system is shown in Figure 5.4 and is made up of three sections: *Random Demodulator*, *Spectral Shaper*, and *Feature Extraction and Classification*. Input signals are simultaneously filtered and sampled at a sub-Nyquist rate in the *Random Demodulator* section using a Markov chain-generated spectrally shaped bipolar pseudorandom sequence $PRS(t)$ generated by the *Spectral Shaper* section. Features are then extracted from the acquired samples and used to detect and identify vehicles in the *Feature Extraction and Classification* section.

In our application, the frequency content that needs to be amplified or attenuated can be determined by examining the average frequency-domain plots of the vehicle classes under consideration. Figure 5.2 shows the average sound signals of passing cars, scooters, and periods without a passing vehicle, sampled for a duration $T_s = 2s$ at a rate $W = 48\,\mathrm{kHz}$. We can see that

between 0-3 kHz, the signals are very difficult to distinguish, and that from 3 kHz onwards, the signals show significantly clearer separation. We refer to this frequency location as the *frequency pivot point*, and in our work we will look to attenuate the sub-3 kHz content to the left of the pivot point, and amplify the rest of the frequency content to the right of the pivot point. As highlighted in Figure 5.4, in our proposed system, the processes that constitute the *Random Demodulator* and *Spectral Shaper* sections are performed on continuous-time analogue signals, and the processes that constitute the *Feature Extraction and Classification* section are performed on discrete-time digital signals. These stages will be examined in more detail in the rest of Section 5.3.

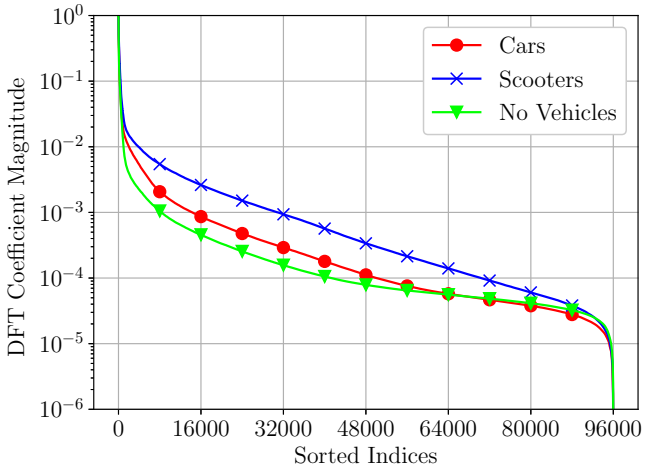


Fig. 5.5: Sorted DFT coefficients $|\alpha_n|$ (rescaled).

Finally, it is necessary to ensure that in our proposed application, the input signals to the RD are sparse or compressible in the domain defined by $\boldsymbol{\Psi}$. A key feature of our proposed system is the simultaneous sampling and filtering performed during the signal acquisition process using the *Spectral Shaper* to match the spectra of the bipolar sequence and input signal; an approach based on the results presented in [69] and [71]. We therefore consider the signals in this chapter to be sparse or compressible in the frequency domain ($\boldsymbol{\Psi}$ = DFT matrix).

Figure 5.5 shows the DFT coefficient distributions of the signals shown in Figure 5.2. We can see that the signals are compressible as their spectra

are clearly dominated by a relatively small number K of high magnitude coefficients.

5.3.1 Random Demodulator

At the heart of our proposed system is the *Random Demodulator* section, which is composed of the following sub-blocks: a signal generator, a mixer, a low-pass filter (LPF), and an analogue-to-digital-converter (ADC). In our proposed C-AVDI system, we include an additional Markov chain block that is used in tandem with the signal generator to spectrally shape the PRS.

An in-depth explanation of the RD's operation was given in Section 2.2.2.

5.3.2 Spectral Shaper

The *Spectral Shaper* section consists of a signal generator and Markov chain block. It is used to produce bipolar sequences with specifically tailored, rather than random, frequency distributions.

The processes and parameters governing the creation of these tailored sequences were covered in detail in Chapter 4.

5.3.3 Feature Extraction and Classification

The *Feature Extraction and Classification* section is composed of a feature extraction block and a classifier block. In the feature extraction block, we extract a set of 5 features from the $\xi[m]$ measurements produced by the RD. We select the 5 most important features out of the 9 used in our preliminary work [39] to use in our C-AVDI system:

- *mean*
- *standard deviation*
- *median*

- *absolute largest value*
- *interquartile range*

In the classifier block, these extracted features are used as inputs to a multiclass classifier to detect and identify passing vehicles. Classification is performed using an RF classifier, chosen for its robustness in the presence of outliers, inherent suitability for multiclass classification, and minimal preprocessing requirements (no input data rescaling required). The RF is implemented using the *scikit-learn* library [78].

5.4 Evaluation

In this section, we evaluate the system described in Section 5.3 by simulating its operation in software. Figure 5.6 shows an overview of the software implementation of the system proposed in Figure 5.4. As previously mentioned, our proposed system operates by taking an audio signal emitted from a passing vehicle as input, sampling it at a sub-Nyquist rate, and outputting a predicted vehicle type. System performance is evaluated using the accuracy metric.

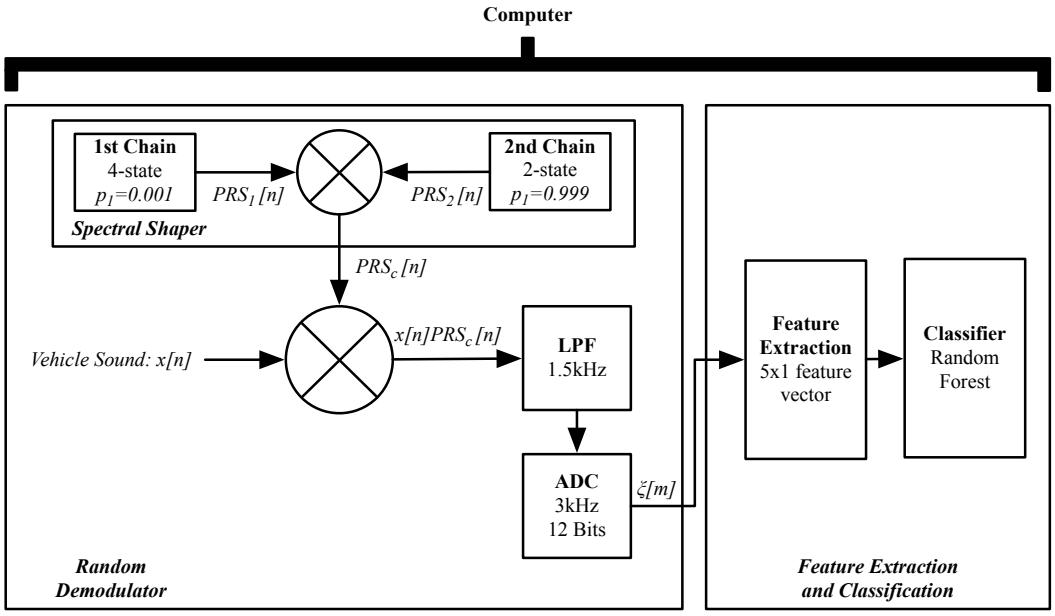


Fig. 5.6: Overview of the C-AVDI software implementation.

Tab. 5.3: Simulated C-AVDI System Parameters

Nyquist Rate W	48 kHz
ADC Rate R	3 kHz
Bit Depth B	12 bits
Signal Length (Time) T_s	$2s$
Signal Length (Samples) N	96000
Signal Sparsity K	400
Compressive Measurements M	6000
1st Chain	4-state
1st Chain Transition Probability p_1	0.001
2nd Chain	2-state
2nd Chain Transition Probability p_2	0.999

5.4.1 System Parameters

We are looking to design a $PRS(t)$ that, when used to demodulate an input signal, will attenuate the sub-3 kHz frequency content to the left of the *frequency pivot point*, and emphasize the frequency content to the right of the pivot point, as shown in Figure 5.2 and Figure 5.3. This same sequence will be used when acquiring data under all three weather conditions, as the location of the pivot point is the same in all three cases. Following the method described in Section 4.3.2, we determine the optimal value of M for our system by first establishing a theoretical lower bound value that is incrementally increased until we obtain the optimal balance between the smallest number of $\xi[m]$ samples and maximum system prediction accuracy. We find this value to be $M = 6000$ and list it, along with the rest of the parameters used in our system simulation, in Table 5.3.

Based on the results obtained in Section 4.3.4, we create a $PRS_C[n] \in \mathbb{R}^N$ from $PRS_1[n] \in \mathbb{R}^N$ a 4-state chain with $p = 0.001$ and $PRS_2[n] \in \mathbb{R}^N$ a 2-state chain with $p = 0.999$. The spectrum of our $PRS_C[n]$ is shown in Figure 5.7.

Our input signal is represented as a discrete version of the analogue input signal $x(t)$, defined as $x[n] \in \mathbb{R}^N$ sampled at rate $W = 48\,\text{kHz}$ during the data

acquisition process. We approximate the sparsity level of $x(t)$ as $K = 400$ by determining the number of DFT coefficient magnitudes shown in Figure 5.5 such that $|\alpha_n| \geq 10^{-1}$, and rounding to a single significant figure.

The anti-aliasing LPF preceding the ADC is a 2nd order Butterworth filter, and the measurements $\xi[m] \in \mathbb{R}^{\mathbb{M}}$ are obtained by sampling and quantizing the combined (using elementwise multiplication) $x[n]PRS_C[n]$ signal at rate $R = 3\,\text{kHz}$ and bit depth $B = 12$ Bits. This corresponds to a reduction in sample rate by a factor of $(\frac{N}{M}) = (\frac{W}{R}) = 16$.

The discrete-time and frequency-domain representations of the average $\xi[m]$ measurements for each class obtained by our simulated system are shown in Figure 5.8.

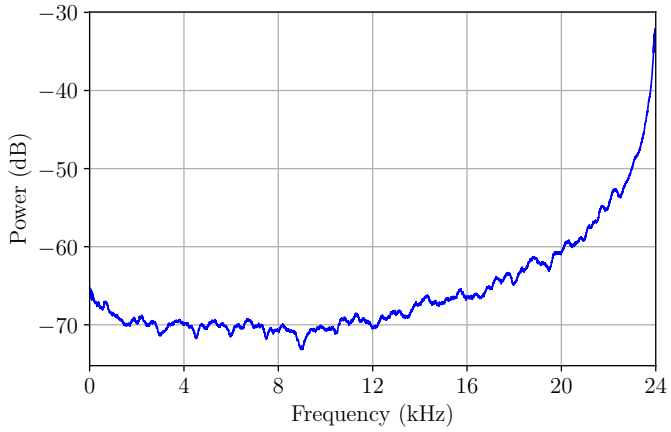
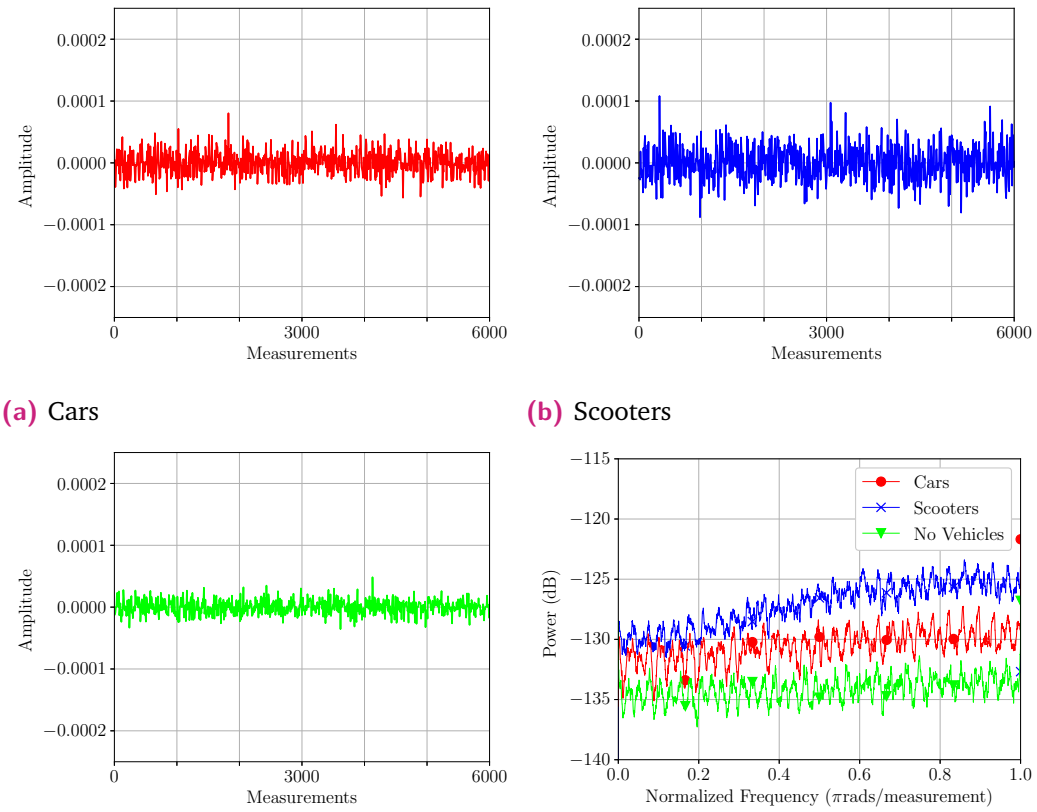


Fig. 5.7: Spectrum of tailored $PRS_C[n]$.

5.4.2 Feature Extraction and Classification

Classification is performed on 14 feature sets, which are obtained by extracting the 5 features outlined in Section 5.3.3 from the $\xi[m]$ measurements of each of the 14 datasets described in Section 5.2. We measure the classification performance of our system using *leave-one-day-out* cross-validation: we set each one of the 14 feature sets (where each set corresponds to a different day) as the testing set, and the combined 13 remaining feature sets act as the training set. This process is performed 14 times in total, with each of the feature sets acting as the testing set in turn. We set the number of trees and



(a) Cars

(b) Scooters

(c) No passing vehicles

(d) Normalised frequency representations

Fig. 5.8: Compressive measurements $\xi[m]$ drawn from the average audio signals of cars, scooters, and periods of time with no passing vehicles, along with their respective normalised frequency representations.

the minimum number of samples per leaf parameters of the RF classifier as $\mathrm{n\_estimators} = 1000$ and $\mathrm{min\_samples\_leaf} = 3$ respectively, with the other parameters left as default. Both the training and testing sets are balanced using random undersampling prior to classification, and the results obtained from each of the 14 runs are averaged to obtain the system accuracy. To account for any potential discrepancies caused by inherent randomness in the classification process, we perform the full classification process 10 times and average the obtained results, resulting in our final confusion matrix shown in Figure 5.9 and the system accuracy scores by weather condition shown in Table 5.4.

The confusion matrix in Figure 5.9 shows that the most prominent misclassification is that of “Car” as “Scooter”. This can be explained by the similar

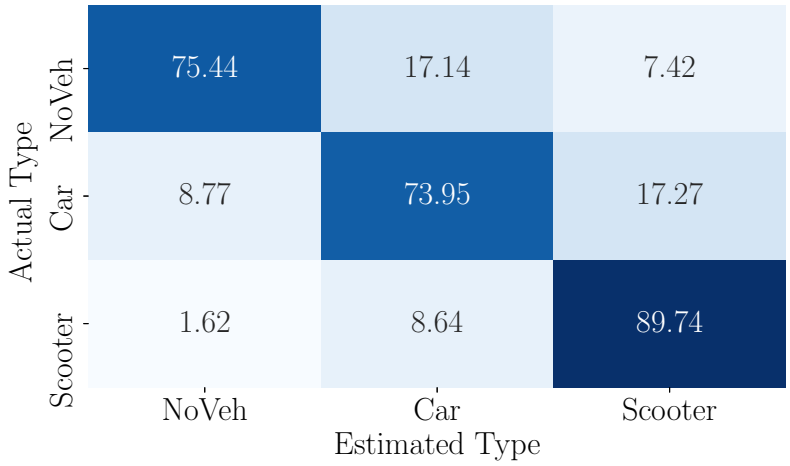


Fig. 5.9: Confusion matrix of C-AVDI's software implementation results. Average accuracy is 80.1%.

Tab. 5.4: System Accuracy by Weather Condition

		All Conditions	Clear	Wind	Rain
Number of Vehicles	Car	1414	623	443	348
	Scooter	777	378	278	121
	NoVeh	2414	1117	751	546
Metric [%]	Accuracy	80.1	84.8	55.2	79.7

variance and amplitude of their $\xi[m]$ samples, which translates to similar *standard deviation* and *interquartile range* features in particular.

This information, along with the results presented in Table 5.4, gives us insight into how to modify the system to improve performance in future evaluations. Most notably, the breakdown of metrics by weather condition sheds light on one of the principal causes of misclassification: adverse weather conditions. The design of a system capable of more effectively mitigating the effects of wind and rain noise on classification accuracy will be the focus of future work.

5.4.3 Benchmarking

We gauge the computational performance of our proposed C-AVDI system by benchmarking it against a simplified single-microphone version of SMBAS, our previous system presented in [40], which we refer to as Mod-SMBAS. For

Tab. 5.5: Benchmarking Results

		C-AVDI	Mod-SMBAS
Time [s]	Extract Features	1.33	3.65
	Classify	0.09	0.28
	Total	1.42	3.93
Metric [%]	Accuracy	80.1	87.3

benchmarking purposes, we change the classifier from SVM to RF, change the input signal from stereo to mono by removing one of the microphones, and remove the signal emphasis process.

Benchmarking is performed by running the systems under test on the same computer under identical conditions and comparing their runtimes. We use a computer equipped with an Intel i9-9900K 16-core CPU @ 3.60 GHz with 64GB of memory running Ubuntu 18.04. As in the previous section, the full process is run 10 times, from which we calculate the respective average runtimes.

The benchmarking results in Table 5.5 show that there is a link between computation time and performance: Mod-SMBAS shows the best accuracy, but is more computationally intensive than C-AVDI by a factor of ≈ 2.8.

5.5 Results and Discussion

We can assess the performance and viability of our C-AVDI system by comparing it to SMBAS, our previous system presented in [40]. While SMBAS presents an accuracy of 95.0%, which is higher than the 80.1% obtained by our C-AVDI system, there are two key differences to consider when comparing these results. The first is the weather conditions in which the systems were tested: SMBAS used only vehicle sounds obtained under clear conditions, and was therefore not tested for robustness to adverse conditions, whereas the C-AVDI system has been tested under different weather conditions as outlined in Table 5.2 and Table 5.4. The second is the computational cost: as discussed in Section 5.4.3 our C-AVDI system runs approximately 2.8 times faster than Mod-SMBAS, the simplified version of our previous system.

Taking these factors into consideration makes the difference in performance between these two systems much less marked, as in clear conditions the

C-AVDI system obtains an accuracy of 84.8% while running approximately 2.8 times faster than Mod-SMBAS which presents an only marginally higher accuracy of 87.3%. The similar accuracy score combined with a faster computation time when compared to a system of similar complexity (mono input signal, limited or no post-sampling processing, use of RF classifier), serves to demonstrate the viability of our proposed C-AVDI system, particularly in applications where low-cost, low-complexity sensing is required. On the other hand, the full implementation of SMBAS is better suited to applications where a high-cost, high-complexity system is required due to its superior accuracy performance.

There is a notable limitation in our proposed system that need to be addressed in any future work: the performance of our proposed C-AVDI system in adverse weather conditions (windy conditions in particular). This limitation can be mitigated by incorporating an additional passive or active noise cancellation stage inspired by our previous work [79] [80], by implementing CSP-based interference removal techniques as presented in [65], or by designing more complex bipolar sequences for more precise filtering during signal acquisition.

The obtained M values listed in Table 4.1 and Table 5.3 would suggest that in our particular C-AVDI application, in which we perform classification without reconstruction, the requirements on the lower bound value of M can be relaxed. Indeed, we can observe an inconsistency between the two values of M with regard to the rest of their respective system parameters. According to Equation 2.9, we would expect the value of M listed in Table 4.1 to be smaller than the value listed in Table 5.3; however, in our case, the opposite is true. Given that the purpose of the system described by the parameters in Table 4.1 is to minimize the RE, and that the purpose of the system described by the parameters in Table 5.3 is to maximize the classification accuracy, their respective values of M would seem to indicate that a smaller number of $\xi[m]$ measurements is required for optimal classification than for optimal reconstruction. A more rigorous investigation of this phenomenon is left as a topic for future research.

Finally, the concepts underpinning C-AVDI presented in this chapter can be used to extend the benefits of sub-Nyquist sampling to a wide variety of

different acoustic (smart device wake-up-word detection) sensing applications, further contributing to the democratization of acoustic sensor-based ML systems.

5.6 Summary

In this chapter, we presented C-AVDI, a method to detect and identify vehicles from their audio signatures at sub-Nyquist rates using features extracted from the compressive measurements obtained by a modified random demodulator. Our system follows the approach described in Chapter 4, and uses Markov chain-generated spectrally shaped bipolar sequences to target a specific frequency band in the input signal during the sampling process itself.

The experimental evaluation of a simulated version of our system under a range of weather conditions produced a classification accuracy of 80.1% for a back-end ADC sample rate of 3 kHz, with a runtime approximately 2.8 times quicker than Mod-SMBAS, a modified version of the frequency-domain feature-based method described in Section 3.2.

We cover the MCU implementation of the C-AVDI system in Chapter 7.

6

C-HAR: Compressive Measurement-Based Human Activity Recognition

6.1 Introduction

In this chapter, we present our compressive measurement-based human activity recognition system (C-HAR), the second of our two proposed compressive measurement-based lightweight classification systems. We begin by describing the human activity dataset used in our approach, before giving an overview of C-HAR's operation, followed by presenting the system's evaluation results.

6.1.1 Background

Lightweight human activity recognition (HAR) is a particular subcategory of HAR which focuses on maximising system efficiency while presenting performance metrics comparable to regular, non-lightweight, alternatives. In our research we are interested in wearable sensor-based lightweight HAR, and we focus on smartwatches in particular as they strike the optimum balance between adoption rate, and computational and sensing capabilities. Indeed, smartwatches are becoming more powerful, slowly making specialized sensors redundant, while also becoming increasingly widespread. Market analysis conducted by [81] shows that the global smartwatch market

volume was 68.59 million units in 2020, and is predicted to reach 230.30 million units by 2026, with the adoption rate growing across all demographics. While smartphones remain more widely used and computationally capable, they suffer from being placed in locations that are far from optimal for HAR, such as in pockets or bags, and on tables.

We examined a large selection of smartwatch-based approaches in Section 3.4.1, but despite being smartwatch-based, not all of them were intrinsically lightweight, as they either required the use of a large amount of data to operate (such as the system proposed in [63] which obtains a large amount of sensor data and sends it via Bluetooth Low Energy (BLE) to a smartphone for off-device processing), or made use of larger-scale more complex classifiers to identify the various activities (such as the approach presented in [64] in which the authors perform classification using large-scale ANNs). It is important to make this distinction, as while some smartwatch-based HAR systems are lightweight, not all of them are, and their design and approach to HAR depends on the requirements of a given application.

In particular, lightweight HAR sees extensive use in medical applications due to two key factors. The first is that low-cost, low-complexity systems can be worn for longer periods of time, increasing the usage rate of the subjects. This is made clear in [9] where the authors found that 27% of participants in a medical trial stopped using their wearable sensing devices due to difficulties ensuring the devices remained powered and operational. The second is that more complex sensor-based systems often transmit information from the sensor or sensors to a central sink node or cloud computing node for processing and storage. This significantly increases the risk of a subject's personal information being intercepted and misused by a third-party company or malicious actor. This also has a significant impact on the adoption rate of wearable HAR devices, as highlighted in [82], where the authors found that a significant set of users in a medical devices trial did not feel comfortable letting their personal information be transmitted to a remote location for processing. Lightweight systems capable of performing processing and classification on-device can help resolve both these limitations.

It follows from the above, that leveraging the inherent dimensionality reduction properties of compressive sensing (CS) to further reduce the amount of data used in lightweight HAR systems can have a clear and quantifiable

benefit on the accessibility, efficiency, and ultimately the adoption of the technology.

6.1.2 Related Work

We examined a selection of more traditional HAR systems and approaches in Section 3.4. In this section we focus on lightweight CS-based HAR related work.

The natural data reduction properties of CS make it particularly suited as a base on which to create a lightweight HAR architecture. As a result, a significant amount of research has been conducted with the aim of improving the efficiency of HAR systems using CS.

The authors of [10] leverage CS to reduce the sampling rates of wearable sensors used in healthcare monitoring applications, in turn lowering the required transmission power between the sensors placed on patients and a central processing computer. Once received, the compressive samples are reconstructed and used to monitor the patients' status. While this system uses CS to acquire input signals at sub-Nyquist rates, it differs majorly from our proposed system as it recovers the original signal from the compressive measurements before performing classification.

In [83], the authors present a compressive measurement-based stroke detection system which bypasses the CS reconstruction process entirely, and performs classification using the information contained on the measurements prior to reconstruction. While this approach is similar to the one put forward in our work, there are two key differences between the two. The first difference is the way in which the compressive-measurements are processed: in our proposed method, the filtering of the input signal occurs during the sampling process, so our compressive measurements are "pre-filtered" rather than being filtered post-sampling. The second difference is the number of class labels: our proposed method performs multiclass classification, and the system outlined in [83] performs binary anomaly detection classification.

The authors of [13] present a low-power compressive sensing-inspired sub-Nyquist sensing device. The system uses Non-Uniform Wavelet Sampling

(NUWS) to obtain a set of features directly from an electrocardiogram (ECG) signal, which are then used to detect cardiac arrhythmia. Again, this approach is similar to C-HAR, however this system does not process or filter the compressive measurements before classification, and similarly to the approach described previously in [83], only performs binary anomaly detection classification.

6.1.3 Goals and Objectives

Our objective is to create a lightweight sub-Nyquist smartwatch-based HAR system. We are seeking to further reduce the cost and complexity compared to existing systems while maintaining a comparable performance. This is achieved using CS to obtain a set of reduced compressive measurements which we use in tandem with a classifier to identify a selection of general human activities. Reducing the number of samples enables us to lower the required system memory by reducing model size and input data storage requirements, and increase device battery life by lowering the computational cost. We use the baseline smartwatch HAR approach outlined in Section 3.4.1 as a starting point from which to create a system which uses spectrally tailored bipolar pseudo-random sequences to simultaneously filter and sample smartwatch sensor readings at sub-Nyquist rates following the approach outlined in Chapter 4.

Our main contributions can be summarised as follows:

- We present C-HAR, a compressive measurement-based HAR system, capable of performing multiclass classification across a whole range of daily human activities while operating at sub-Nyquist rates and obtaining accuracy comparable to existing approaches.
- We simultaneously sample and filter sensor readings using spectrally shaped bipolar sequences generated from a pair of Markov chains. We tailor the choice of bipolar sequence to both the sensor, and the sensor axis. Adapting our sampling and filtering strategy in this manner improves the classification accuracy of our system.

- Demonstrate the viability of C-HAR as a lightweight sensing architecture by benchmarking our proposed system against a selection of baseline approaches.

In this chapter, we focus on creating the C-HAR architecture by simulating and optimising the various parameters involved in the system design process in software. The MCU implementation of the feature extraction and classification processes of our proposed system is described in-depth in Chapter 7.

6.2 Experimental Data

6.2.1 WISDM Dataset

The data used in this proposed system is taken from the WISDM dataset [51]. The dataset is made up of data sampled at a rate of 20 Hz from the accelerometers and gyroscopes of smartphones and smartwatches worn and carried by a set of 51 subjects, who each performed one of 18 different activities for a duration of 3 minutes, resulting in a total of 2754 minutes of data. The activities can be split into 3 categories, which are shown in Table 6.1.

Tab. 6.1: Activities by Category

Category	Non-Hand-Oriented	Hand-Oriented (General)	Hand-Oriented (Eating)
Activities	Walking	Dribbling (Basketball)	Pasta
	Jogging	Ball Catching	Soup
	Climbing Stairs	Typing	Sandwich
	Standing	Writing	Chips
	Ball Kicking	Brushing Teeth	Drinking
	Sitting	Clapping	
		Folding Clothes	

6.2.2 Preprocessing

The WISDM dataset is subject to a number of preprocessing operations before being used in the proposed system. The first step is to discard all the data collected by smartphones, retaining only the data collected by smartwatches. As alluded to previously, a smartwatch's placement on a subject's wrist

with three degrees of freedom makes the resultant sensor readings more representative of the activities performed by the subject, and thus particularly suited to HAR applications. By contrast, a smartphone is usually placed in a pocket or in a bag, making the obtained sensor readings considerably less representative of any activities performed by the subject. The second step is to remove the data of any subjects who did not perform all 18 activities, bringing the number of subjects down from 51 to 43. The final step is to split the sensor data by subject and by activity into non-overlapping windows of length t_w . The WISDM dataset comes with a range of features extracted from 10s windows of the raw time-series data, thus for the sake of consistency and future comparison of results, we set $t_w = 10s$. A detailed breakdown of the total number of segments per subject and activity are shown in Table 6.2 for the accelerometer data, and in Table 6.3 for the gyroscope data.

The smartwatch used to collect the data was the LG G Watch running Android Wear 1.5 [1].

6.3 System Overview

An overview of our proposed C-HAR system is shown in Figure 6.1 and is made up of three sections: *Random Demodulator*, *Spectral Shaper*, and *Feature Extraction and Classification*. Input signals drawn from the accelerometer and gyroscope sensor readings of a smartwatch are simultaneously filtered and sampled at a sub-Nyquist rate in the *Random Demodulator* section using a Markov chain-generated spectrally shaped bipolar pseudorandom sequence $PRS(t)$ generated by the *Spectral Shaper* section. Features are then extracted from the acquired samples and used to detect and identify a range of daily human activities in the *Feature Extraction and Classification* section. Some details relating to the operation of specific sections and blocks of the system have been covered in more detail in previous chapters. We will indicate in the text when this is the case, and refer to the relevant chapter or section.

Given that a key feature of our proposed system involves matching the spectra of our input and spreading signals, it is important to visualise the frequency

[1] https://www.gsmarena.com/lg_g_watch_w100-7718.php

Tab. 6.2: Accelerometer Data by Subject and Activity

Subject	Walking	Jogging	Stairs	Sitting	Standing	Typing	Brushing	Soup	Chips	Pasta	Drinking	Sandwich	Kicking	Catching	Dribbling	Writing	Clapping	Folding	Total
1600	11	11	11	11	13	13	11	11	11	11	11	11	11	11	11	11	11	10	201
1601	14	14	14	14	14	14	15	15	14	15	14	14	15	15	15	14	14	14	258
1602	11	11	11	11	10	11	11	11	11	11	11	11	11	11	11	11	11	11	197
1603	15	15	15	15	14	14	15	15	14	14	15	14	15	15	14	14	15	15	263
1604	11	11	11	11	11	9	11	11	11	11	11	11	11	11	11	11	11	11	196
1605	15	15	15	14	14	15	14	15	14	15	14	14	15	15	15	15	15	15	264
1606	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	10	197
1607	15	15	15	14	15	15	14	15	14	15	15	14	15	15	15	14	14	14	263
1608	11	10	11	11	11	11	11	10	11	6	11	11	11	11	11	11	11	11	191
1609	11	11	11	11	11	11	11	11	11	11	7	11	11	10	11	11	11	11	193
1610	11	11	11	11	11	11	11	11	11	11	11	11	9	11	11	11	11	11	196
1611	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	198
1612	11	10	11	11	11	11	11	11	11	8	11	10	11	11	11	11	11	11	193
1613	14	14	15	15	14	14	15	14	15	15	14	15	15	15	14	14	14	12	258
1614	13	13	10	13	12	13	13	12	13	13	13	13	13	13	13	13	13	12	228
1615	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	9	11	11	196
1617	11	11	11	11	11	11	11	11	11	11	10	11	11	11	8	11	10	10	192
1619	14	14	14	13	13	13	13	13	13	14	13	14	14	14	11	13	10	13	236
1620	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	8	11	11	195
1621	14	13	14	7	10	14	13	14	14	13	13	13	13	13	13	14	13	13	231
1622	11	10	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	197
1623	11	11	11	10	11	11	11	11	11	11	11	11	11	11	11	11	11	11	197
1624	11	10	11	11	11	11	11	11	11	11	11	11	10	10	11	5	11	8	186
1625	11	10	9	10	11	11	11	10	11	11	10	10	11	7	11	11	11	11	187
1626	11	11	11	10	11	11	11	11	11	11	7	11	10	10	10	11	11	11	190
1627	11	9	10	11	11	11	11	11	11	11	11	11	10	10	11	11	11	10	192
1628	11	11	11	11	4	11	11	11	11	11	10	11	11	11	11	11	11	11	190
1630	11	10	11	10	11	11	11	11	11	11	11	11	11	10	10	10	11	11	193
1631	11	10	11	11	11	11	11	10	11	11	11	11	10	11	10	11	11	11	194
1632	11	11	11	11	11	11	10	11	11	10	11	11	11	11	11	11	11	11	196
1633	11	11	11	11	11	11	11	11	11	11	11	11	11	11	10	11	11	11	197
1634	11	11	11	11	11	11	11	8	11	11	11	11	11	10	10	11	11	11	193
1635	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	12	11	11	199
1636	11	10	11	11	11	11	11	11	11	11	11	11	10	10	10	11	11	11	194
1641	14	15	14	14	15	14	15	15	15	15	14	15	14	14	14	15	15	15	262
1643	15	15	14	14	15	15	14	15	14	15	12	15	15	13	14	15	15	14	259
1644	14	14	15	15	15	15	14	15	15	15	12	15	14	10	10	15	15	15	253
1645	14	14	8	15	15	14	15	14	15	15	15	12	15	15	14	15	15	15	255
1646	14	14	15	15	15	15	15	14	14	14	15	14	13	14	14	15	15	15	260
1647	12	11	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	215
1648	15	14	15	15	14	15	15	15	15	15	14	15	14	14	14	15	15	15	264
1649	12	12	13	12	12	13	12	13	13	13	13	13	12	10	12	13	13	13	224
1650	10	15	15	15	15	15	15	10	15	14	15	14	14	15	15	15	15	15	257

Tab. 6.3: Gyroscope Data by Subject and Activity

Subject	Walking	Jogging	Stairs	Sitting	Standing	Typing	Brushing	Soup	Chips	Pasta	Drinking	Sandwich	Kicking	Catching	Dribbling	Writing	Clapping	Folding	**Total**
1600	11	11	11	11	12	12	11	11	11	11	11	11	11	11	11	11	10	11	199
1601	14	14	14	14	14	14	14	14	14	14	14	15	14	14	14	14	14	14	253
1602	11	11	11	11	10	11	11	11	11	11	11	11	11	11	11	11	11	11	197
1603	15	14	14	15	14	14	14	14	14	13	14	14	14	14	14	14	14	15	254
1604	11	11	11	11	11	9	11	11	11	11	11	11	11	11	11	11	11	11	196
1605	15	15	14	14	14	14	14	14	13	15	14	14	15	14	14	15	15	14	257
1606	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	10	11	197
1607	15	14	14	14	14	14	14	14	14	14	14	14	14	15	14	14	14	13	253
1608	11	10	11	11	11	11	11	10	11	6	11	11	11	11	11	11	11	11	191
1609	11	11	11	11	11	11	11	11	11	11	7	11	10	11	11	11	11	11	193
1610	11	11	11	11	11	11	11	11	11	11	11	9	11	11	10	11	11	11	195
1611	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	198
1612	11	11	11	11	11	11	11	11	11	8	11	11	11	11	11	11	11	10	194
1613	14	14	14	14	14	14	14	14	14	14	14	15	15	14	14	15	12	14	253
1614	13	13	10	13	12	12	12	12	13	13	13	13	13	13	13	13	12	13	226
1615	11	11	11	11	11	11	11	11	11	11	11	11	11	11	9	11	11	11	196
1617	11	11	11	11	11	11	11	11	11	11	10	11	11	8	11	10	10	11	192
1619	14	14	13	13	13	13	13	13	13	14	13	13	14	11	13	10	13	14	234
1620	11	11	11	11	11	11	11	11	11	10	11	11	11	11	8	11	11	11	194
1621	14	13	14	7	10	14	13	14	14	13	13	14	13	13	14	13	13	13	232
1622	11	10	11	11	11	11	11	11	12	11	11	11	11	11	11	11	11	11	198
1623	11	11	11	10	11	11	11	11	11	11	11	11	11	11	11	11	11	11	197
1624	11	10	11	11	11	11	11	11	11	11	11	10	10	11	5	11	8	11	186
1625	11	10	9	10	11	11	11	10	11	11	10	11	7	11	11	11	11	10	187
1626	11	11	11	10	11	11	11	11	11	11	7	10	10	10	11	11	11	11	190
1627	11	9	10	11	11	11	11	11	11	11	11	10	10	11	11	11	10	11	192
1628	11	10	11	11	4	11	11	11	11	11	10	11	11	11	11	11	11	11	189
1630	11	10	11	10	11	11	11	11	11	11	11	11	10	10	10	11	11	11	193
1631	11	10	11	11	11	11	11	10	11	11	11	10	10	10	11	11	11	11	193
1632	11	11	11	11	11	11	10	11	11	11	11	11	11	11	11	11	11	11	197
1633	11	11	11	11	11	11	11	11	11	11	11	11	11	10	11	11	11	11	197
1634	11	11	11	11	11	11	11	8	11	11	11	11	10	10	11	11	11	11	193
1635	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	198
1636	11	10	11	11	11	11	11	11	11	11	11	10	10	10	10	11	11	11	193
1641	14	15	14	14	15	14	15	15	15	15	14	14	14	14	15	15	15	15	262
1643	15	15	14	14	15	15	14	15	14	15	12	15	13	14	15	15	14	15	259
1644	14	14	15	15	15	15	14	15	15	15	12	14	10	10	14	15	15	15	252
1645	14	14	9	15	15	14	15	14	15	15	15	15	15	14	15	15	14	12	255
1646	14	14	15	15	15	15	14	14	14	14	15	13	14	14	15	15	15	14	259
1647	12	11	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	12	215
1648	15	14	15	15	14	15	15	15	15	15	14	13	14	13	15	15	15	15	262
1649	12	12	13	12	12	13	12	13	13	13	13	13	10	12	13	13	13	13	225
1650	11	15	15	15	15	15	15	11	15	14	15	14	15	15	15	15	15	14	259

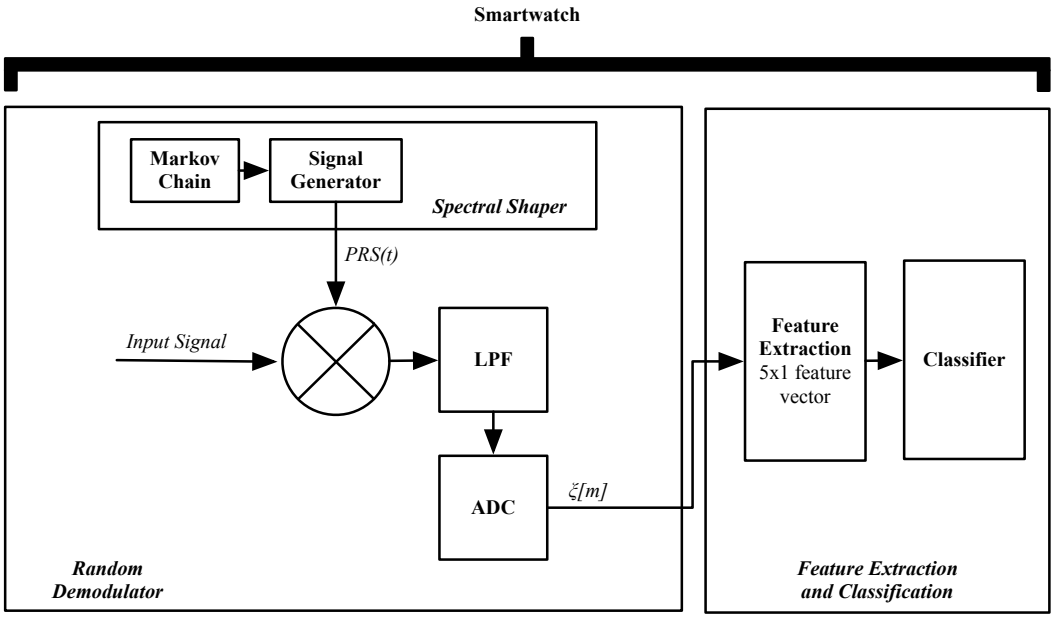


Fig. 6.1: C-HAR system overview: an input signal $x(t)$ is simultaneously sampled and filtered during the signal acquisition process using a spectrally shaped bipolar pseudorandom sequence $PRS(t)$ generated by the *Spectral Shaper*.

domain representations of the signals we are looking to acquire. The average frequency domain plots of the accelerometer and gyroscope readings by axis and activity are shown in Figure 6.2 and Figure 6.3 respectively.

We can see that the spectra vary quite significantly depending on the sensor, activity, and axis. Whereas in Chapter 5 it was possible to create a single $PRS(t)$ tailored to the 3 vehicle classes, the variety of different frequency spectra in the current C-HAR application make this impossible. To address this problem, we propose a selection of different operating modes for our system, with each mode offering presenting a different approach to signal acquisition and feature extraction. The final system will be able to select the operating mode which yields the best performance for a given set of input signals. Our C-HAR system presents four different operating modes which all use the same basic system blocks, but in a different manner and with different parameters. The four different approaches are shown in Figure 6.4, and are explained in further detail in Section 6.3.1.

It is necessary to ensure that in our proposed application, the input signals to the random demodulator (RD) are sparse or compressible in the domain defined by $\mathbf{\Psi}$, where $\mathbf{\Psi}$ = DFT matrix as in Chapter 5.

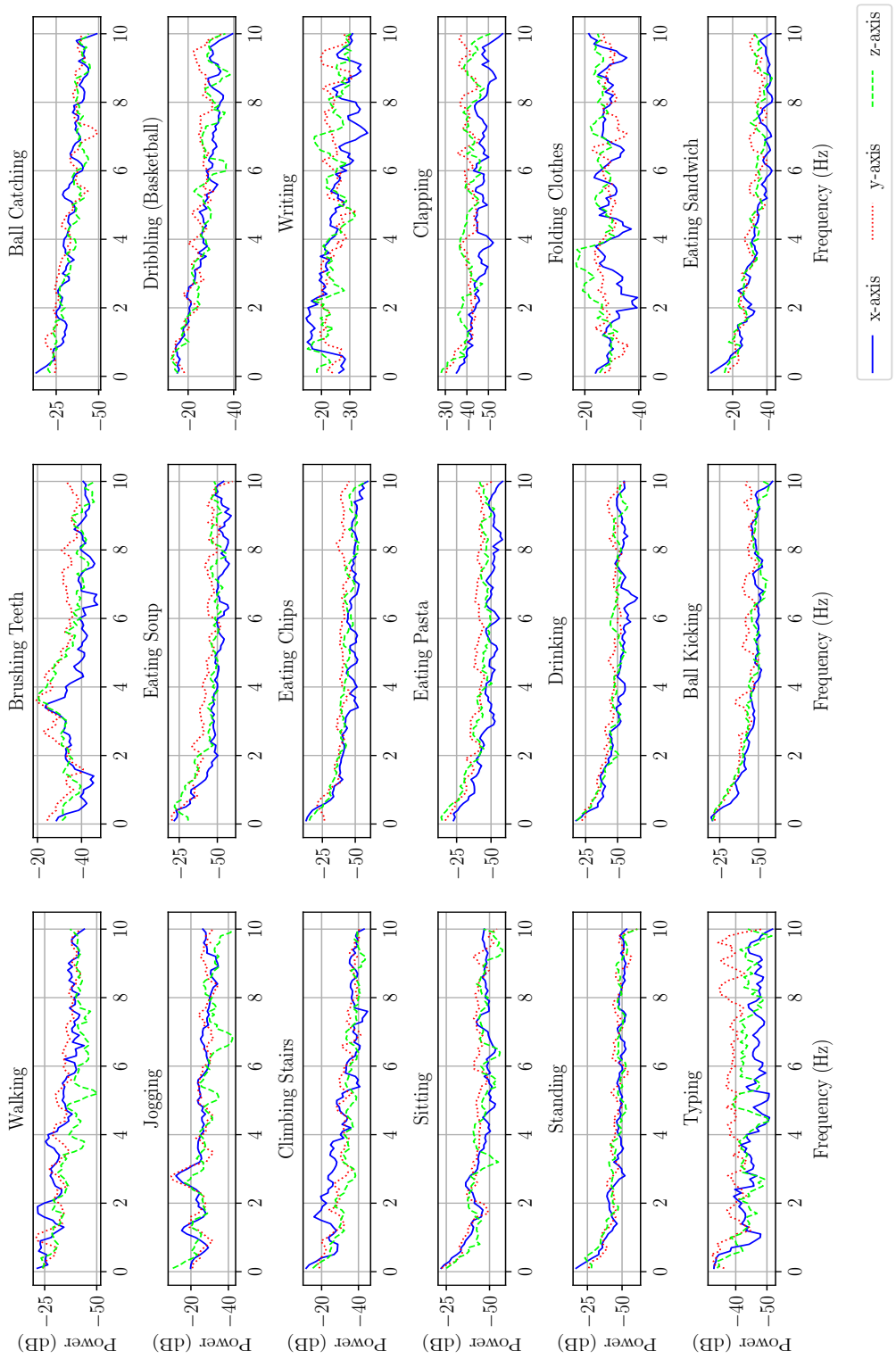


Fig. 6.2: Average frequency domain plots of the accelerometer readings by axis and activity.

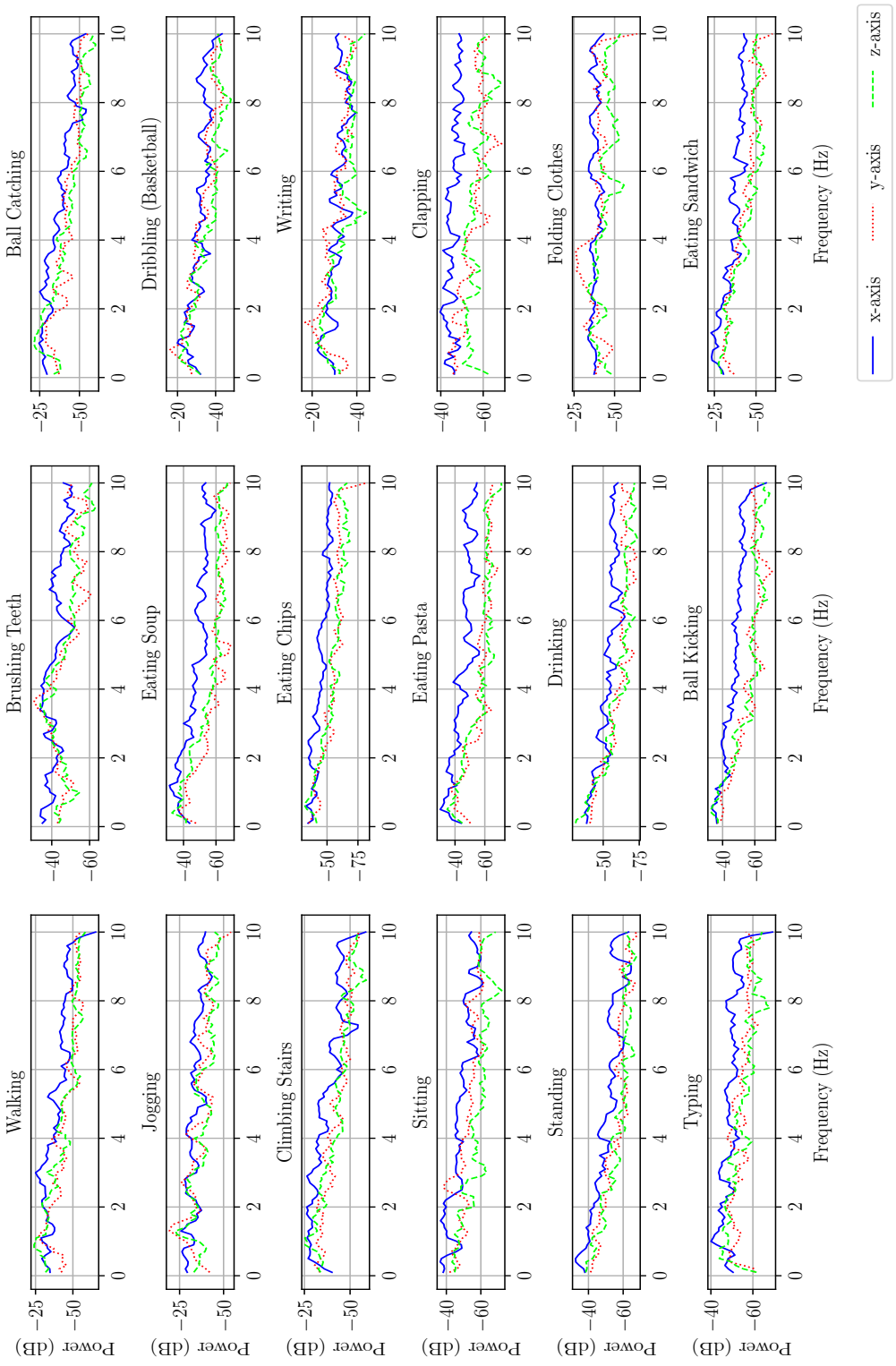


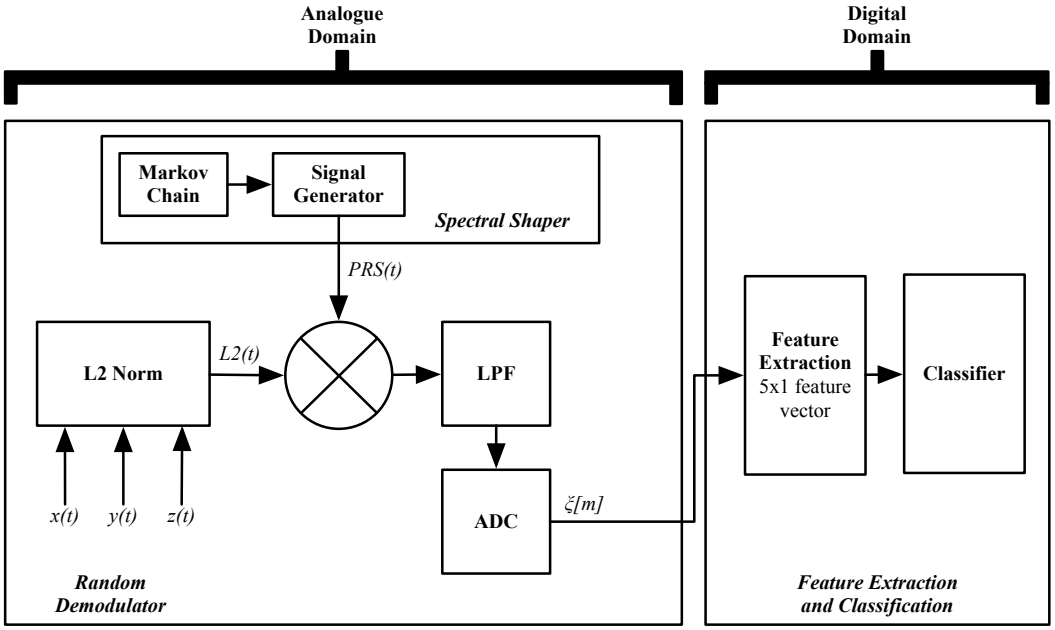
Fig. 6.3: Average frequency domain plots of the gyroscope readings by axis and activity

Tab. 6.4: Compressibility Ratios of Sensor Signals by Activity and Axis

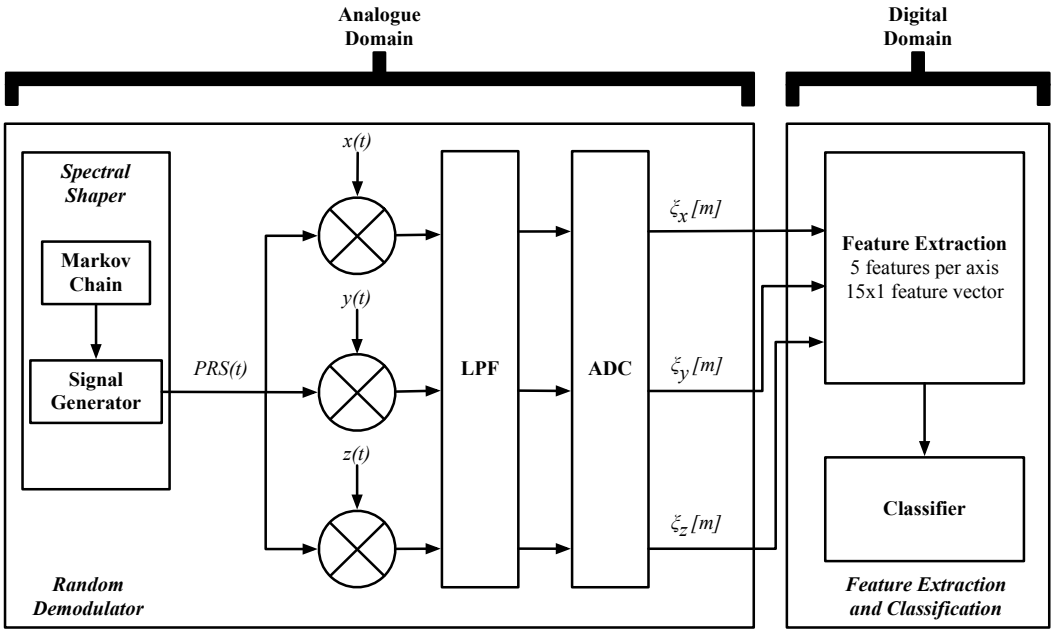
Sensor	Accelerometer			Gyroscope		
Activities	x-axis	y-axis	z-axis	x-axis	y-axis	z-axis
Walking	0.35	0.23	0.31	0.33	0.05	0.08
Jogging	0.28	0.1	0.15	0.67	0.14	0.22
Climbing Stairs	0.21	0.36	0.35	0.5	0.35	0.07
Sitting	0.11	0.14	0.16	0.47	0.28	0.25
Standing	0.02	0.19	0.09	0.23	0.29	0.2
Typing	0.5	0.68	0.69	0.73	0.64	0.5
Brushing Teeth	0.34	0.38	0.33	0.59	0.13	0.36
Soup	0.06	0.07	0.08	0.19	0.08	0.16
Chips	0.05	0.12	0.06	0.45	0.25	0.13
Pasta	0.06	0.09	0.08	0.29	0.21	0.2
Drinking	0.03	0.03	0.06	0.28	0.2	0.09
Ball Kicking	0.06	0.08	0.11	0.27	0.23	0.16
Ball Catching	0.15	0.34	0.28	0.42	0.3	0.21
Dribbling	0.45	0.54	0.21	0.46	0.2	0.27
Writing	0.55	0.8	0.65	0.46	0.34	0.47
Clapping	0.71	0.59	0.13	0.66	0.4	0.69
Folding Clothes	0.79	0.84	0.58	0.8	0.41	0.62
Sandwich	0.04	0.44	0.17	0.36	0.36	0.28

We define the compressibility ratio of a signal as $\frac{K}{N}$, and list the compressibility ratios of each activity for all axes and both sensors in Table 6.4. The compressibility ratio serves two purposes: first it gives as idea of the structure of the signals under test, and their suitability for CS applications. We can see that apart from some outliers activities such as "Writing", "Folding", and "Clapping", most sensor readings can be treated as compressible. Second, we can use the compressibility ratio to derive M, the lower bound on the number of required compressive measurements. We calculate values of M_a and M_g heuristically for the accelerometer and gyroscope respectively by taking the average of the compressibility ratios of each sensor and using the inverse of the result in Equation 2.9. We obtain values of $M_a = 71.14$ and $M_g = 72.23$. We saw in Chapter 5 that a smaller number of $\xi[m]$ measurements is required for optimal classification than for optimal reconstruction. This allows us to relax the lower bound requirements on M in our system. For the sake of simplicity, we pick M such that the result of $\frac{N}{M}$ is an integer, which gives us $M = M_a = M_g = 50$ and $\frac{N}{M} = 4$.

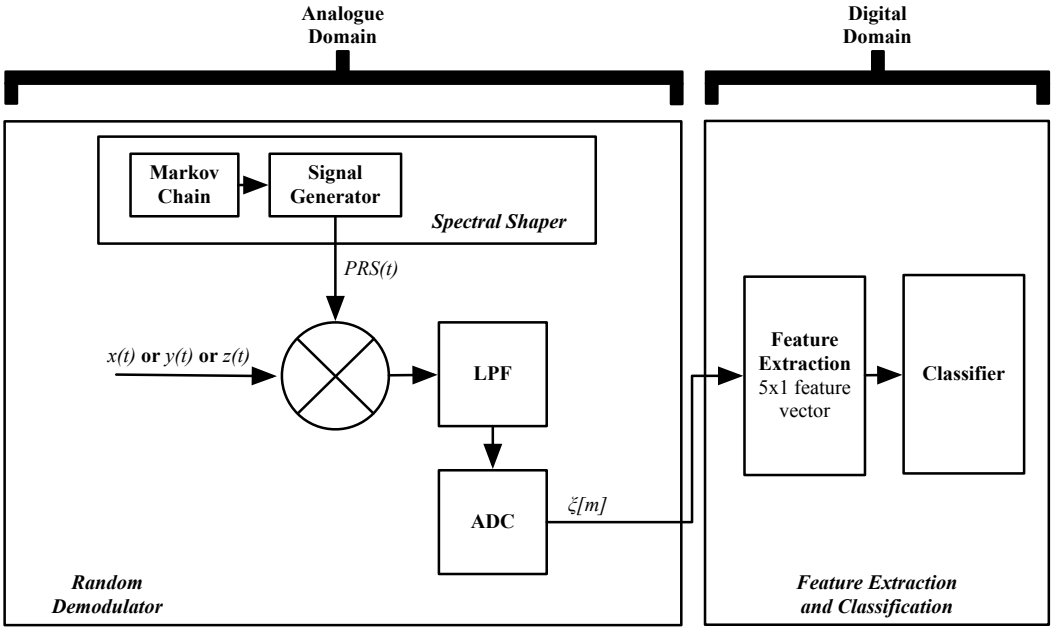
Finally, because the sampling rates of the two sensors are not synchronised, in the rest of this chapter we perform evaluation using the accelerometer and gyroscope readings separately, rather than a single, combined sensor reading.



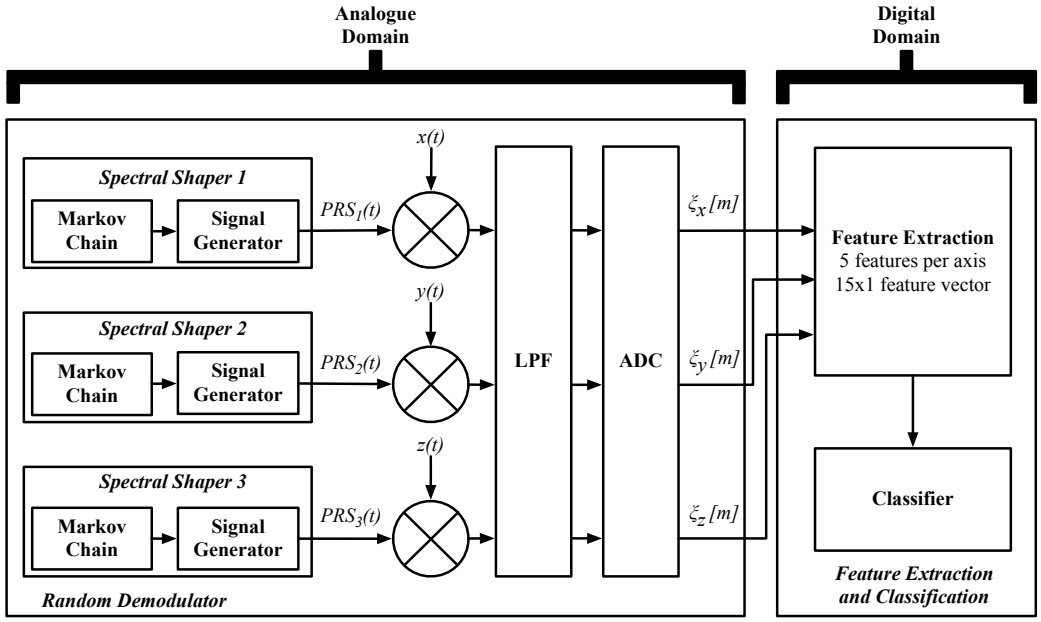
(a) *Normed* axis operating mode



(b) *Combined* axis operating mode



(c) *Individual* axis operating mode



(d) *Matched* axis operating mode

Fig. 6.4: System overview of C-HAR system in *normed*, *combined*, *individual*, and *matched* operating modes.

6.3.1 Operating Modes

In our proposed system, we put forward four different operating modes (referred to *normed axis*, *combined axis*, *individual axis*, and *matched axis* respectively) used to perform C-HAR based on both accelerometer and gyroscope sensor data:

- The *normed axis* operating mode consists of taking the L2 norm of the three axis-data and using the result as input to the rest of the system. This input signal is then combined with a single $PRS(t)$, before being low-pass filtered, and sampled at a rate R. A set of 5 features is extracted from the $\xi[m]$ measurements for activity classification.
- The *combined axis* operating mode consists of combining the sensor readings of each of the three axes with a common $PRS(t)$. These three demodulated signals are then low-pass filtered, and sampled at a rate R. A set of 15 features (5 from each of the 3 sets of $\xi[m]$ measurements obtained from the modulated signals) is extracted for activity classification.
- The *individual axis* operating mode consists of combining the sensor readings of each one of the three axes with a $PRS(t)$, in turn. This demodulated signal is then low-pass filtered, and sampled at a rate R. A set of 5 features is extracted from the $\xi[m]$ measurements for activity classification. This process is performed a total of three times, once for each of three sensor axes.
- The *matched axis* operating mode consists of combining the sensor readings of each of the three axes by matching them with the most suitable $PRS(t)$ based on the results of the *individual axis* operating mode. These three demodulated signals are then low-pass filtered, and sampled at a rate R. A set of 15 features (5 from each of the 3 sets of $\xi[m]$ measurements obtained from the modulated signals) is extracted for activity classification.

6.3.2 Random Demodulator

The *Random Demodulator* section is central to our system, and is made up of a signal generator, a mixer, a low-pass filter (LPF), and an analogue-to-digital-converter (ADC). In our proposed C-HAR system, we include an additional Markov chain block that is used in tandem with the signal generator to spectrally shape the PRS. The RD is designed using the same parameters in each of the operating modes, and acquires a single input signal in the *normed axis* and *individual axis* operating modes, and 3 input signals in the *combined axis* and *matched axis* operating modes.

An in-depth explanation of the RD's operation was given in Section 2.2.2

6.3.3 Spectral Shaper

The *Spectral Shaper* consists of a signal generator and Markov chain block. It is used to produce bipolar sequences with specifically tailored, rather than random, frequency distributions.

In our application, we are faced with 18 different activities recorded with 2 different sensors over 3 axes, which makes for a total of 108 different signals whose frequency domain representations are shown in Figure 6.2 and Figure 6.3. This makes it very difficult to visually determine what frequency content needs to be amplified or attenuated to ensure the best possible classification accuracy over all activities. Thus, rather than trying to visually match the activities with a suitable $PRS(t)$, we create a set of four filter sequences based on the results of Chapter 4: a low-pass sequence, a band-pass sequence, a high-pass sequence, and a broadband sequence referred to as PRS_{LP}, PRS_{BP}, PRS_{HP}, and PRS_{BB} respectively. The spectra of these sequences are shown in Figure 6.5, and the parameters used to generate them are shown in Table 6.5.

The processes governing the creation of these tailored sequences were covered in detail in Chapter 4.

Tab. 6.5: Filter $PRS(t)$ Parameters

PRS	Number of Chains	p_1	p_2	1^{st} Chain Length	2^{nd} Chain Length
PRS_{LP}	Single	0.85	-	2	-
PRS_{BP}	Dual	0.9	0.9	2	4
PRS_{HP}	Single	0.1	-	4	-
PRS_{BB}	Single	0.5	-	2	-

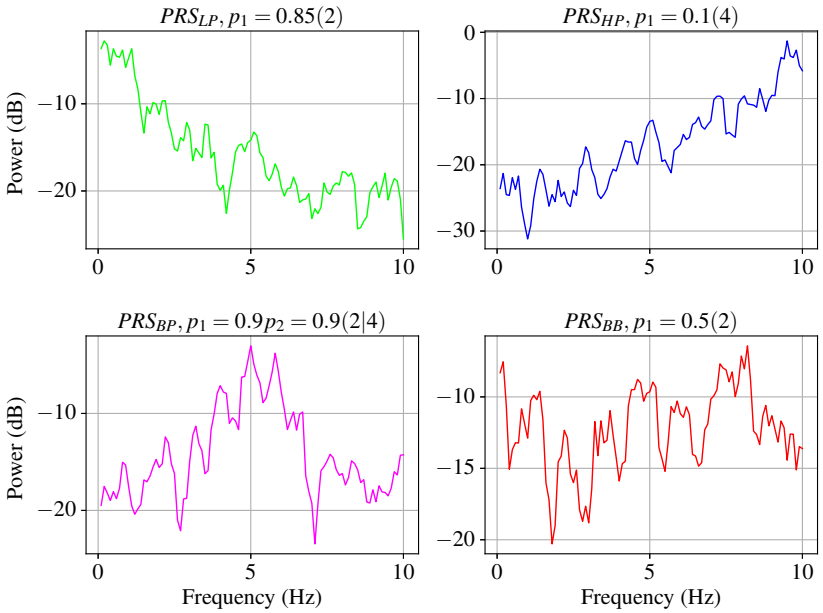


Fig. 6.5: Filter $PRS(t)$ signals. The generation parameters and chain lengths of each signal are listed above each subplot.

6.3.4 Feature Extraction and Classification

The *Feature Extraction and Classification* section is composed of a feature extraction block and a classifier block. We extract a set of 5 features from each set of $\xi[m]$ measurements produced by the RD, which makes a total of 5 features for the *normed axis* and *individual axis* operating modes, and a total of 15 (3×5) for the *combined axis* and *matched axis* operating modes.

The 5 features extracted from each set of $\xi[m]$ measurements are:

- *mean*
- *standard deviation*

- *median*
- *absolute largest value*
- *interquartile range*

In the classifier block, these extracted features are used as inputs to a multiclass classifier to detect and identify passing vehicles. Classification is performed using a random forest (RF) classifier, a common choice in HAR applications due to its suitability for multiclass classification, minimal preprocessing requirements (no input data rescaling required), and good outlier tolerance. The RF is implemented using the *scikit-learn* library [78].

6.4 Evaluation

Initial attempts at performing 18-class classification showed it to be an impossible task under our self-imposed lightweight system constraints. Accurately classifying 18 different activities involves extensive input signal preprocessing or the use of a consequentially larger classifier, both of which require significantly more computational power than what is found in current lightweight approaches. Indeed, the lightweight HAR systems presented in section 6.1.2 only perform binary anomaly detection. Given that the activities present in our used dataset are split into 3-categories, we evaluate our C-HAR system using successive 3-class classification as this enables us to effectively and fairly evaluate system performance under the current computational constraints.

Thus, evaluation is performed by drawing one activity from each of the three activity categories (Non-hand-oriented, hand-oriented (general), hand-oriented (eating)) in turn, performing multiclass classification on every possible 3-activity combination, for a total of 210 combinations. This is repeated for each operating mode. For the *normed axis*, *combined axis* and *individual axis* modes the optimal filtering $PRS(t)$ and results are obtained iteratively. The optimal parameters and results of the *matched axis* mode are obtained based on the *individual axis* mode results. The filter $PRS(t)$ parameters are the same as defined previously in Table 6.5, the other parameters are taken from the WISDM dataset, and the value of M was determined

previously in Section 6.3. The sample rate $R = 5$ Hz represents a reduction of the Nyquist rate $W = 20$ Hz by a factor of 4. Performance is evaluated using the accuracy metric.

Overviews for the both the accelerometer and gyroscope *matched axis* operating mode systems are shown in Figure 6.6. All parameters are summarized in Table 6.6.

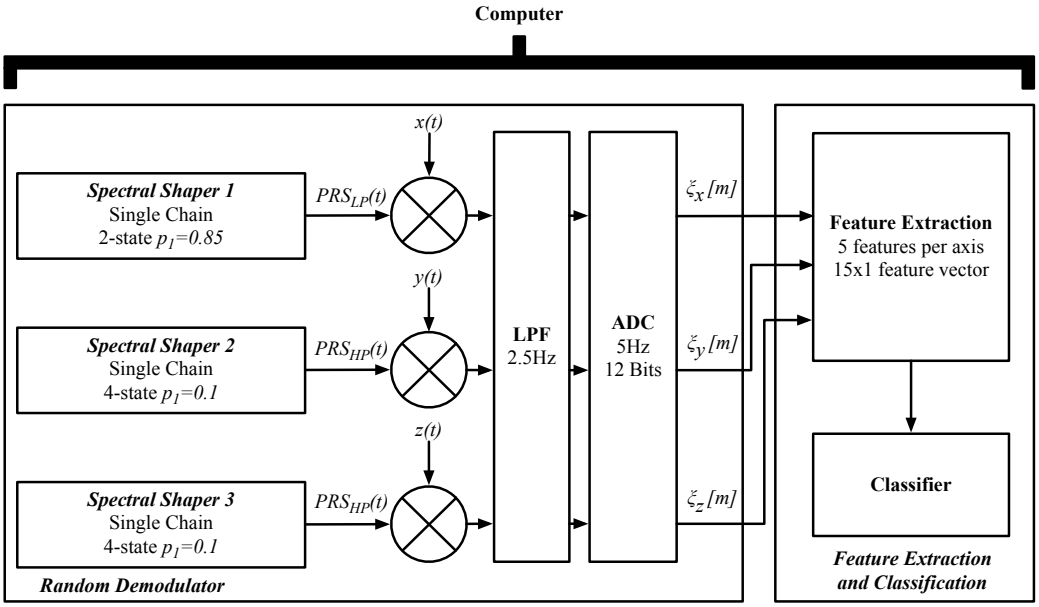
Tab. 6.6: Simulated C-HAR System Parameters

Nyquist Rate W	20 Hz
ADC Rate R	5 Hz
Bit Depth B	12 bits
Signal Length (Time) T_s	$10s$
Signal Length (Samples) N	200
Signal Sparsity K	*See Table 6.4*
Compressive Measurements M	50
1st Chain	*See Table 6.5*
1st Chain Transition Probability p_1	*See Table 6.5*
2nd Chain	*See Table 6.5*
2nd Chain Transition Probability p_2	*See Table 6.5*

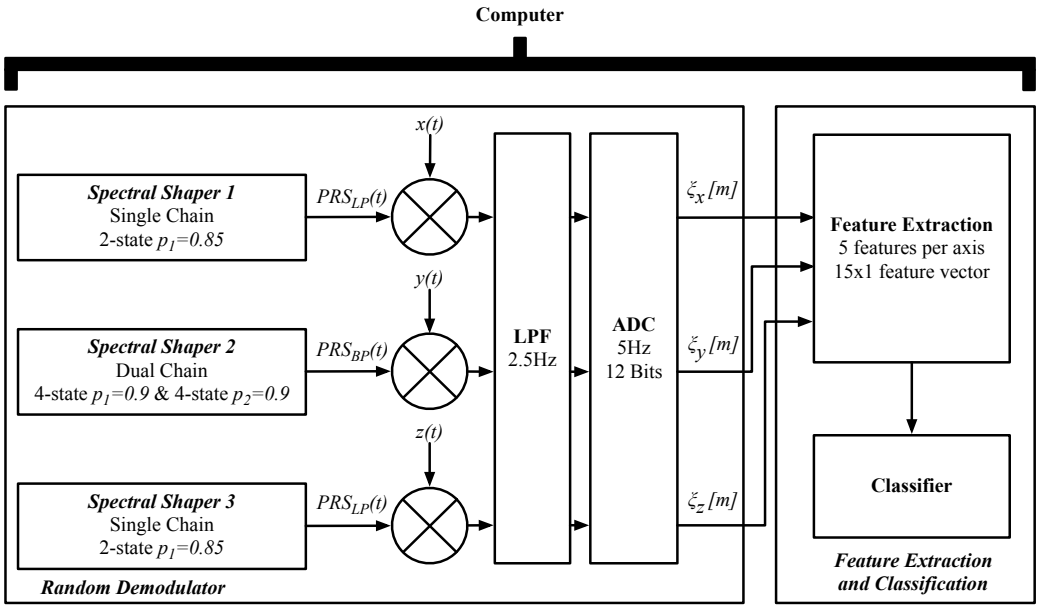
6.4.1 Feature Extraction and Classification

We measure the classification performance of C-HAR using *leave-one-subject-out* cross-validation: we set each one of the 43 feature sets (where each set corresponds to a different test subject) as the testing set, and the combined 43 remaining feature sets act as the training set. This process is performed 43 times in total, with each of the feature sets acting as the testing set in turn.

We set the number of trees and the minimum number of samples per leaf parameters of the RF classifier as $\mathrm{n\_estimators} = 1000$ and $\mathrm{min\_samples\_leaf} = 3$ respectively, with the other parameters left as default. Both the training and testing sets are balanced using random undersampling prior to classification, and the results obtained from each of the 43 runs are averaged



(a) *Matched axis* operating mode using the accelerometer



(b) *Matched axis* operating mode using the gyroscope

Fig. 6.6: System overview of final C-HAR system in *matched axis* operating mode using the accelerometer, and the gyroscope.

to obtain the system accuracy. To account for any potential discrepancies caused by inherent randomness in the classification process, we perform the full classification process 10 times and average the obtained results.

6.4.2 Results

Tab. 6.7: System Accuracy by Operating Mode

(a) Accelerometer

Operating Mode	Normed Axis	Combined Axis	Individual Axis		
			x-axis	y-axis	z-axis
Number of Features	5	15	5	5	5
Accuracy PRS_{LP}	95.4%	98.3%	95.6%	93.7%	93.3%
Accuracy PRS_{BP}	95.3%	98.3%	95.1%	93.5%	93.4%
Accuracy PRS_{HP}	94.3%	98.2%	95.5%	93.8%	94.1%
Accuracy PRS_{BB}	95.1%	98.1%	94.6%	92.9%	92.5%
Accuracy $NOCS$	96.6%	98.9%	97.2%	95.6%	95.4%

(b) Gyroscope

Operating Mode	Normed Axis	Combined Axis	Individual Axis		
			x-axis	y-axis	z-axis
Number of Features	5	15	5	5	5
Accuracy PRS_{LP}	94.6%	97.3%	93.5%	94.5%	94.2%
Accuracy PRS_{BP}	94.5%	97.4%	93.3%	94.9%	94.1%
Accuracy PRS_{HP}	93.8%	97.1%	93.0%	94.3%	94.0%
Accuracy PRS_{BB}	94.4%	97.1%	93.1%	94.6%	94.0%
Accuracy $NOCS$	95.0%	98.0%	94.0%	95.6%	95.3%

The final classification accuracy results of each of the 210 3-class groups, obtained for the *normed axis*, *combined axis* and *individual axis* operating modes, for each filter $PRS(t)$ are averaged and summarised for both sensors in Table 6.7. The final line of both tables, *NOCS*, shows the accuracy obtained using a non-compressive measurement-based approach, i.e: extracting the same set of features from $N = 200$ unfiltered samples obtained by sampling at the Nyquist rate W.

From Table 6.7 we can obtain information regarding the best performing $PRS(t)$ sequence for each axis for both sensors. We use this information to design the final *matched axis* operating mode, which extracts the following 15 features:

- mean
- standard deviation
- median
- absolute largest value
- interquartile range

for

- x-axis
- y-axis
- z-axis

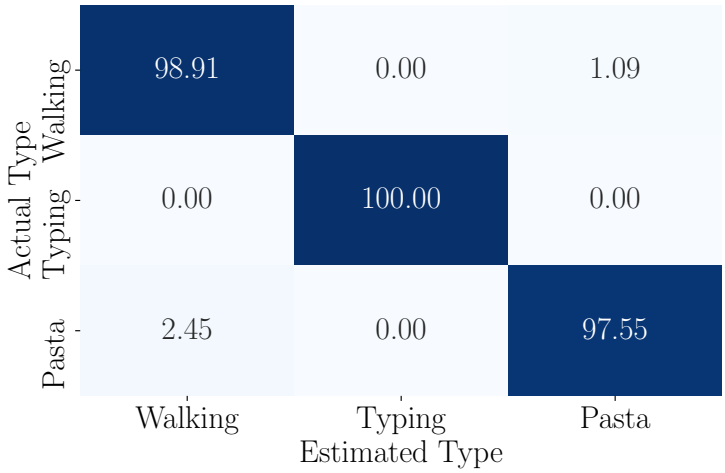
and whose parameters and results are summarised in Table 6.8.

Finally, for the sake of completeness, we include in Figure 6.7 a confusion matrix for one of the 3-activity selections. We choose the 3-activity combination to be as representative as possible of the system. This is done by selecting activities with an average compressibility ratio (as to not inflate performance results by actively cherry-picking particularly sparse activities) from each of the 3 activity categories as defined in Table 6.1: we choose "Walking", "Typing", and "Pasta". We then perform classification using the processes described above using a reduced dataset containing only these three activities.

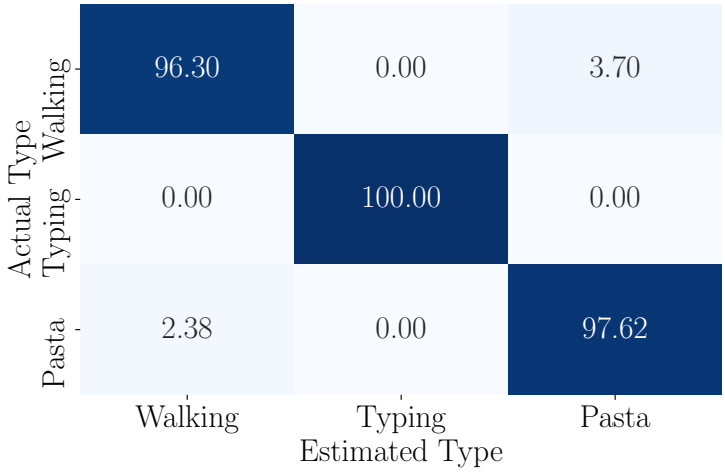
Looking at Figure 6.7, we can see that both sensors are able to perfectly recognise "Typing", but occasionally misclassify "Pasta" and "Walking". This can be explained by the hand and wrist movement during these activities: when a subject is typing, their wrists remain static, or only move very slightly, whereas when walking, the hands and wrists swing with the arms, and when eating pasta they perform a range of different hand- and wrist-based movements. We speculate that the similarities in range and frequency of hand and wrist movement between "Pasta" and "Walking" when compared to "Typing" are the cause of misclassification.

Tab. 6.8: Matched Operating Mode Results

Sensor	x-axis $PRS(t)$	y-axis $PRS(t)$	z-axis $PRS(t)$	Number of Features	Accuracy
Accelerometer	PRS_{LP}	PRS_{HP}	PRS_{HP}	15	98.5%
Gyroscope	PRS_{LP}	PRS_{BP}	PRS_{LP}	15	97.6%



(a) Accelerometer-based system results



(b) Gyroscope-based system results

Fig. 6.7: Confusion matrices for accelerometer-based system results and gyroscope-based system results. Average accuracy is 98.8% and 98.0%, respectively.

Tab. 6.9: Benchmarking Results

System	Extract Features	Classify	Total	Sample Rate	Number of Features	Accuracy
Accelerometer, *Normed Axis*	0.55s	59.3s	59.9s	5Hz	5	95.4%
Gyroscope, *Normed Axis*	0.56s	58.1s	58.7s	5Hz	5	94.6%
Accelerometer, *Combined Axis*	0.75s	62.3s	63.1s	5Hz	15	98.3%
Gyroscope, *Combined Axis*	0.75s	63.7s	64.5s	5Hz	15	97.4%
Accelerometer, *Individual Axis*	0.48s	59.2s	59.7s	5Hz	5	95.6%
Gyroscope, *Individual Axis*	0.48s	59.5s	60.0s	5Hz	5	94.9%
Accelerometer, *Matched Axis*	0.75s	61.2s	62.0s	5Hz	15	98.5%
Gyroscope, *Matched Axis*	0.75s	63.4s	64.2s	5Hz	15	97.6%
Accelerometer, *Baseline*	0.75s	60.1s	60.9s	20Hz	15	98.9%
Gyroscope, *Baseline*	0.75s	62.7s	63.5s	20Hz	15	98.0%

6.4.3 Benchmarking

The most straightforward way to determine the viability of our proposed C-HAR system is to benchmark both the gyroscope- and accelerometer-based systems in each of the four operating modes against baseline versions of the systems. These baseline, non-compressive measurement-based systems operate by sampling each of the 3 axes at the Nyquist rate, extracting 5 features from each axis and combining them into the set of 15 features outlined in Section 6.4.2.

Benchmarking is performed by running the systems under test on the same computer under identical conditions and comparing their runtimes. We use a computer equipped with an Intel i7-11700K 8-core CPU @ 3.60 GHz with 64GB of memory running Ubuntu 20.04. As in the previous section, the full process is run 10 times, from which we calculate the respective average runtimes. The benchmarking results are shown in Table 6.9. We can see that the biggest contributor to computational time is the number of features. Indeed, the operating modes which use 15 features all present comparable total runtimes of 63.0 seconds on average, and the operating modes using 5 features all present comparable total runtimes of 59.6 seconds on average, with no obvious relation between the runtime and sample rate or sensor type.

We can also compare the accuracy performance of C-HAR to other published results, such as those examined in this chapter's literature review. In our work, the setup presenting the best accuracy for the lowest sample rate is the accelerometer-based *matched axis* system, which obtains an accuracy of 98.5% for a sample rate of 5 Hz. This can be compared to the system

in [83] which obtains an accuracy of 91.0% for a sample rate of 10 Hz, and [13] which obtains an accuracy of 98.9% for a wavelet generator clock rate of 360Hz (for the sake of fair comparison, however, it is important to note that the authors of this paper focus on reducing power consumption rather than sample rate, and manage to reduce power consumption by a factor of 15 compared to similar existing ECG anomaly detection systems). From these results, we can infer that C-HAR performs similarly to the two other lightweight C-HAR systems, however it is important to bear in mind the differences in the signals obtained and identified: C-HAR looks to acquire and classify relatively low-noise embedded sensor signals, whilst the other two systems are looking to obtain and identify significantly noisier ECG-signals. It is reasonable to assume that the relative noise levels of the signals, along with other factors, have an effect on the accuracy performance of the different systems.

6.5 Results and Discussion

We have seen that while the exact final accuracy and computational runtime vary depending on the chosen sensor and operating mode, overall, C-HAR presents improved performance metrics when compared to the traditional Nyquist rate baseline approaches. Indeed, our best performing *matched axis* operating mode setups for both sensors obtain better accuracy than their respective baseline approaches (if only marginally), for a sample rate 4 times lower, and comparable runtimes.

While these results are encouraging, to properly gauge the viability of C-HAR, it is necessary to compare it to other existing lightweight HAR systems. We find that its accuracy is marginally higher, and sample rate marginally lower than the two comparable existing works outlined in Section 6.4.3, the differences in approach, sensors, and crucially, application, make it difficult to directly compare the systems.

From this we can conclude further study of C-HAR is required to confirm its status as a viable compressive measurement-based lightweight HAR architecture. We can outline two potential ways in which this can be achieved. The first approach involves refining our current smartwatch-based system

to further reduce the sampling rate, or more realistically, the computational runtime, thereby enabling C-HAR to stand alone as a high-accuracy, low-complexity HAR system. The second approach involves implementing the underlying RD- and operating mode-based framework proposed in this chapter in an ECG signal-based HAR system, and comparing the results with the previously outlined related work. This more direct comparison to existing state-of-art systems would help confirm the viability of C-HAR, particularly in the field of medical HAR.

6.6 Summary

In this chapter, we presented C-HAR, a method to identify daily human activities from the sensor readings of a smartwatch's onboard accelerometer and gyroscope. Our system follows the approach described in Chapter 4, and uses Markov chain-generated spectrally shaped bipolar sequences to target a specific frequency band in the input signal during the sampling process itself.

We find that with the optimal setup and parameters, C-HAR obtains accuracies of 98.5% and 97.6% for the accelerometer-based and gyroscope based systems respectively, for a back-end ADC sample rate of 5 Hz, which is 4 times smaller the back-end ADC sample rate of 20 Hz used in the baseline approaches.

We cover the MCU implementation of the C-HAR system in Chapter 7.

7 Microcontroller Implementation

7.1 Introduction

The main goal of this research is to create a lightweight classification architecture capable of operating on resource-limited devices. Thus, it is crucial to demonstrate that the systems described in the two previous chapters can be implemented on microcontrollers (MCUs). In this chapter, we present the MCU implementations of our compressive measurement-based acoustic vehicle detection and identification system (C-AVDI) and compressive measurement-based human activity recognition system (C-HAR). We begin by explaining our general approach, before describing the two systems and presenting their respective evaluation results.

7.1.1 Background

The MCU implementation of a classification system can be considered as the litmus test for determining whether it is lightweight or not. Indeed, while there exists a range of theoretical systems presenting lightweight classification pipelines, only a small proportion of these approaches actually have existing MCU implementations. The biggest design constraint encountered when creating a classification model for deployment on an MCU is the limited amount of available memory, which places limits on the both the number and length of extracted feature vectors used as inputs to the model, and on the size of the model itself. As a result, we can determine if a model is truly lightweight if it can operate under these constraints.

An MCU implementation brings about a number of significant benefits to a system's form factor, battery life, and cost. A small form factor ensures that an acoustic vehicle detection and identification (AVDI) system is easy to deploy in a variety of environments, and that a human activity recognition (HAR) device is comfortable to wear with minimal impingement to a user's movements. A long battery life ensures that the system can function reliably over the required operational period (this can range from a full working day, to several weeks or months depending on the application and environment). A low cost makes the designed system more readily accessible to a wider range of individuals, communities, and countries, further democratising the use of machine learning- (ML) based systems and their associated benefits.

More generally, the concept of MCU-based ML is particularly relevant when taken in the current context, outlined in the introductory section of our work in Chapter 1. Creating lightweight classification models whose size is stringently limited by the MCUs memory stands as a stark contrast to the ever bigger, more costly, and more energy intensive models seen in both industry and academic research. Taking into account the potential for a system to be deployed on an MCU during the conception phase of an ML product, can help frame and consider the environmental and economical implications of a system, leading to a more efficient and effective final product.

7.1.2 Related Work

Given the relatively small amount of existing work related to MCU-based AVDI, we also examine non-vehicle-oriented MCU-based acoustic detection and identification systems.

The authors of [47] propose an ultra-low power vehicle detector (ULP-VD) capable of detecting passing vehicles with minimal computational cost using the discrete wavelet transform (DWT) in tandem with logistic regression (LR) to detect, but not identify, the presence of passing vehicles. In [11], the authors present an environment monitoring system which uses FFTs and a hidden Markov model (HMM) to detect noises in a woodland environment, which runs on an MCU. The system samples signals at the Nyquist rate, from which features are extracted and used as inputs to a selection binary anomaly

detection classifiers. The authors of [84] present a lightweight environmental sound classification approach for resource constrained MCUs. Using TensorFlow for Microcontrollers[1] they manage to fit a multiclass classification model onto an MCU and achieve an accuracy of 71.0%.

On the other hand, MCU-based HAR applications are more common given the inherent form factor constraints imposed by some smaller wearable HAR devices. In [85] the authors propose a low-power waist-worn HAR device which uses linear discriminant analysis (LDA) for dimensionality reduction on the input samples, and a support vector machine (SVM) classifier to identify a selection of activities. While the system obtains a multiclass accuracy of 98.0%, it requires a large external power source to function making it uncomfortable to wear for long periods of time, and its placement on a subject's waist limits the activities it can detect to standing, walking and running. The authors of [58] present an activity recognition framework which makes use of traditional accelerometer and gyroscope sensor information in conjunction with a wearable stretch sensor placed on a subject's leg to obtain data from multiple sensing modalities. This sensor information is presented to an MCU-based deep neural network (DNN) whose performance can be improved through user feedback. This online learning-capable framework obtains a multiclass accuracy of 94.8%, with user-provided feedback improving accuracy by up to 40% in some cases. This information however, is provided through a smartphone application which also calculates and transmits the necessary weight and bias changes of the DNN, and so it can not be said the entire system back-end is implemented on an MCU. In contrast to the systems outlined above which all use supervised learning methods to perform classification, the authors of [86] propose an unsupervised learning approach using self-organizing maps. The system obtains an unsupervised accuracy of 84.0%, compared to a benchmark supervised accuracy of 92.0% obtained using a convolutional neural network (CNN).

The work described above can be broadly separated into two categories: application-focussed and process-focussed. The first category describes work such as [11] which focuses on sensing in remote environments, where MCUs are an integral part of an overall solution, but are not the primary focus of the research. The second category includes work such as [86] which

[1] https://www.tensorflow.org/lite/microcontrollers

focusses on developing an unsupervised learning pipeline for MCUs, where the MCUs themselves are the main focus of research. Our work falls under the first category, where MCUs are an integral part of our proposed C-AVDI and C-HAR architectures, but not the primary focus.

7.1.3 Goals and Objectives

So far, we have presented in our work two systems divided into a clear front-end section (*Random Demodulator* and *Spectral Shaper* sections) and back-end section (*Feature Extraction and Classification* sections). While the work presented in Chapter 5 and Chapter 6 explored in detail the operation of the entirety of the two systems, and the work presented in Chapter 4 focussed on an integral part of the systems' front-end, this chapter will focus on the systems' back-end.

Our main contributions can be summarised as follows:

- We further demonstrate the viability of C-AVDI and C-HAR as lightweight alternatives to existing AVDI and HAR systems by successfully implementing them on MCUs.
- We examine the limitations faced by the MCU implementations of the system and how to address them in future iterations.

The aim of this chapter is to implement the systems' backend on an MCU, proving the viability of our proposed approaches. We do not focus on the front-end as there are existing hardware implementations of the random demodulator (RD) (most recently in [87]), and creating a custom hardware front-end including both the *Random Demodulator* and *Spectral Shaper* sections is outside the scope of our work.

7.2 General Approach

The first step in the MCU implementation of a system is to select a suitable device. Table 7.1 shows the specifications of some commonly used MCUs.

Tab. 7.1: Specifications of Commonly Used Microcontrollers and Development Boards

MCU	Volatile Memory	Non-Volatile Memory	ADC Resolution	Power Consumption	Unit Price
Teensy 4.1[2] (iMX RT1062[3])	1024 kB	7936 kB	12 bits	10.0 mW at 24 MHz	$28.60
LaunchPad[4] (MSP-EXP430FR599[5])	8 kB	256 kB	12 bits	5.9 mW at 16 MHz	$16.99
Arduino Uno[6] (ATMega328P[7])	2 kB	32 kB	10 bits	26.0 mW at 8 MHz	$25.30
PIC18F452[8]	8 kB	32 kB	10 bits	8.0 mW at 4 MHz	$8.69

[2] https://www.pjrc.com/store/teensy41.html
[3] https://www.embeddedartists.com/products/imx-rt1062-oem/ based on iMX RT1060 https://www.nxp.com/docs/en/application-note/AN12245.pdf
[4] https://www.ti.com/lit/pdf/slau678
[5] https://www.ti.com/lit/ds/slase54d/slase54d.pdf?ts=1626677655975
[6] https://www.farnell.com/datasheets/1682209.pdf
[7] http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-PA-168A-PA-328-P-DS-DS40002061A.pdf
[8] https://ww1.microchip.com/downloads/en/DeviceDoc/39564c.pdf

We choose the Teensy 4.1 board for our implementation as it has the best cost-to-memory ratio, as well as the most total available memory.

The compressive measurements $\xi[m]$ are generated in the manner described in Section 5.4 and Section 6.4, and are saved on an SD card from which they are sequentially loaded into the MCU for feature extraction. The previously trained classifiers used in C-AVDI and C-HAR are ported from their software implementations onto the MCU using the *emlearn* library [88].

7.3 C-AVDI

7.3.1 System Overview

Figure 7.1 shows the MCU implementation of our proposed C-AVDI system. The *Random Demodulator*, and *Spectral Shaper* sections are implemented in software, and the *Feature Extraction and Classification* section is implemented on the MCU.

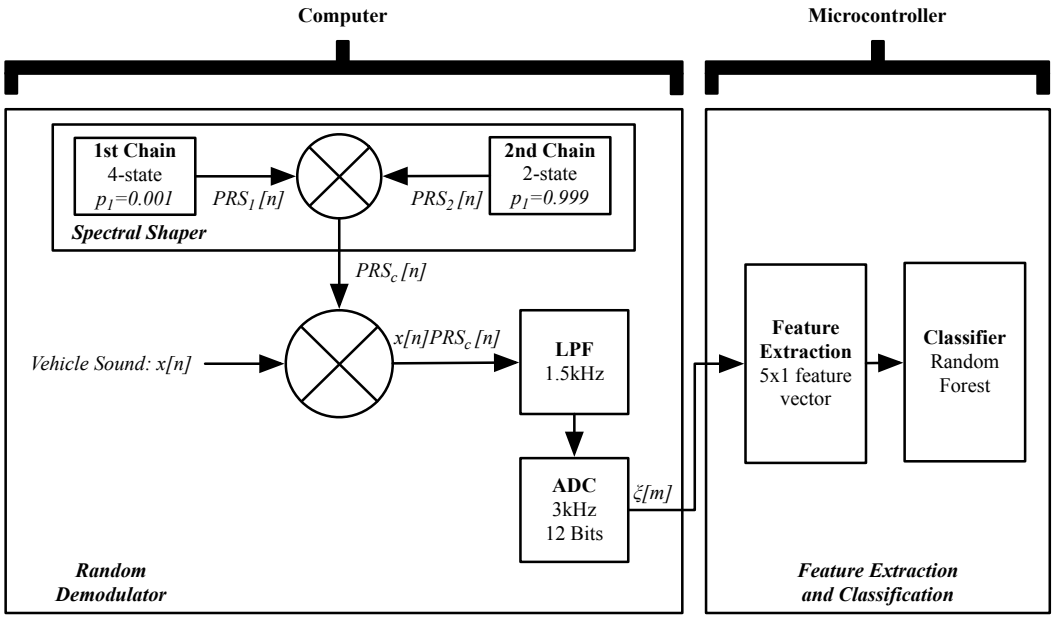


Fig. 7.1: Overview of our C-AVDI system's MCU implementation.

Both the dataset and system parameters used in this MCU implementation are the same as used in the software implementation. The dataset and parameters are displayed in Table 5.2 and Table 5.3 respectively.

7.3.2 Feature Extraction and Classification

After loading the $\xi[m]$ measurements onto the MCU, we extract the same following five features as in our simulated system:

- *mean*
- *standard deviation*
- *median*
- *absolute largest value*
- *interquartile range*

We confirm that the features extracted by the MCU are the same as the features extracted in software by outputting the relevant features to the Teensy

IDE's serial monitor and comparing them to the feature values generated in software.

The classifier used is a smaller ported version of the random forest (RF) classifier trained in the system's software implementation. We set the number of trees and the minimum number of samples per leaf parameters of the RF classifier as $\mathrm{n\_estimators} = 100$ and $\mathrm{min\_samples\_leaf} = 3$ respectively, with the other parameters left as default.

7.3.3 Evaluation and Discussion

Figure 7.2 shows the resulting confusion matrix. We obtain an accuracy of 71.8%, which is lower than the 80.1% accuracy obtained by the simulated system in Section 6.4.2. Given that the features extracted from $\xi[m]$ in the *Feature Extraction and Classification* section in both the software and MCU implementations of our system are identical, we can easily infer that the difference in accuracy between both implementations is due to the difference in size and complexity between the ported and original versions of the classifier models. Indeed, the limited memory of the MCU restricts the number of trees in the RF classifiers to 100, compared to the 1000 used in the software implementation, while also truncating the size of the various coefficients, weights and parameters used in the ported version of the RF classifier, leading to a drop in classification accuracy.

7.4 C-HAR

7.4.1 System Overview

Figure 7.3 shows both the accelerometer and gyroscope variations in *matched axis* operating mode of the MCU implementation of our proposed C-HAR system. In both cases the *Random Demodulator*, and *Spectral Shaper* sections are implemented in software, and the *Feature Extraction and Classification* section is implemented on the MCU.

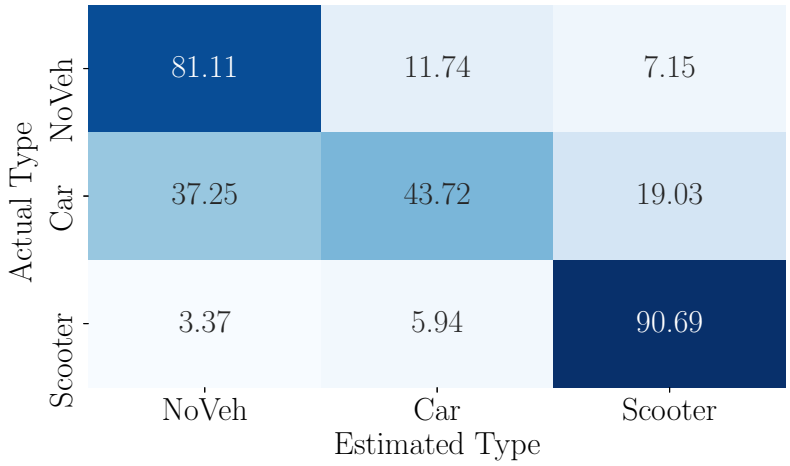
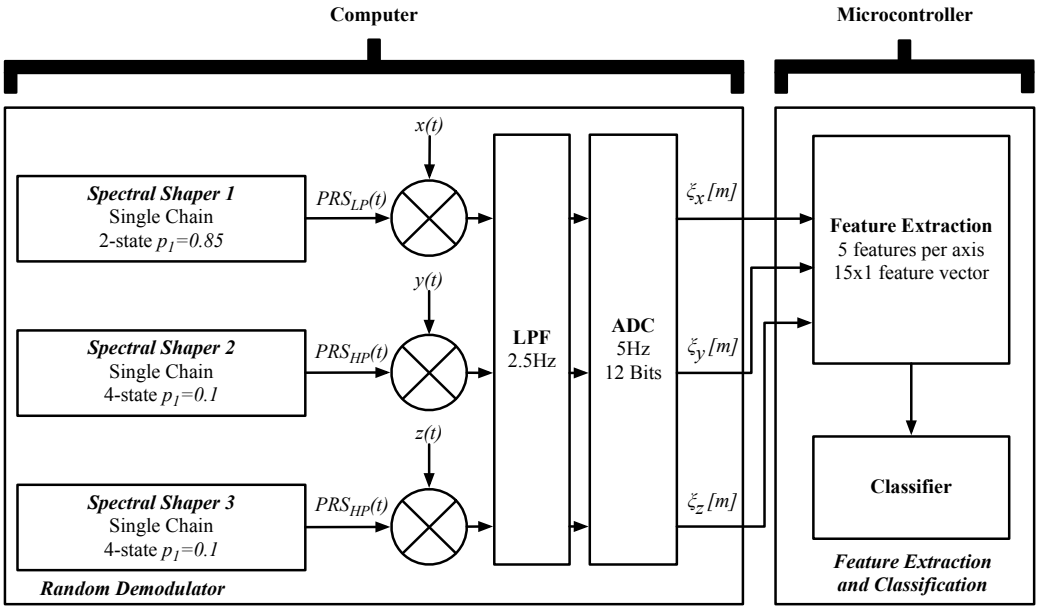


Fig. 7.2: Confusion matrix of the C-AVDI system's MCU implementation. Average accuracy is 71.8%.

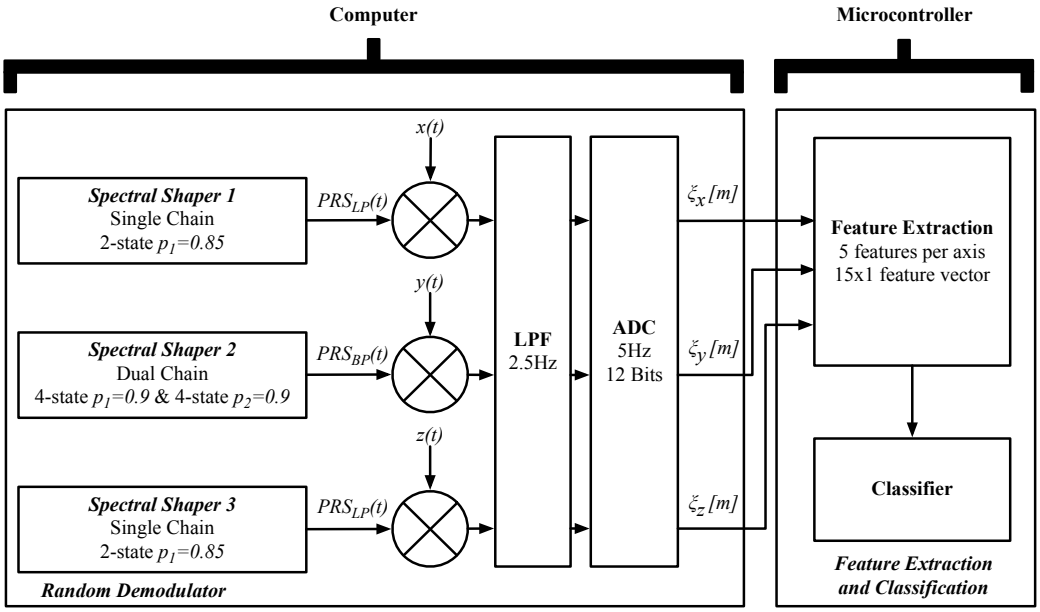
Tab. 7.2: C-HAR MCU Implementation Dataset

	Accelerometer				Gyroscope			
Subject	Walking	Typing	Pasta	Total	Walking	Typing	Pasta	Total
1600	11	13	11	35	11	12	11	34
1602	11	11	15	37	11	11	11	33
1604	11	9	11	31	11	9	11	31
1609	11	11	11	33	11	11	11	33
1610	11	11	11	33	11	11	11	33
1617	11	11	11	33	11	11	11	33
1631	11	11	11	33	10	11	11	32
1634	11	15	11	37	11	11	11	33
1636	11	11	11	33	11	11	11	33
1641	14	14	15	43	14	14	15	43
1645	11	14	15	40	14	14	15	43
1648	15	15	15	45	15	13	15	43
1649	12	13	13	38	12	15	13	40
1650	10	15	14	39	11	15	14	40

The system parameters used in this MCU implementation are the same as used in the software implementation and are displayed in Table 6.6. The dataset used is a reduced dataset using 14 of the original 43 subjects chosen at random, and only a single 3-activity combination, rather than the 210 different combinations used in the software simulation. We choose the same 3-activity combination as in Section 6.4.2 ("Walking", "Typing", and "Eating Pasta") as the average nature of the compressibility ratios of the 3 activities in question make them representative of the system as a whole. The resulting dataset can be seen in Figure 7.2.



(a) *Matched axis* operating mode using the accelerometer



(b) *Matched axis* operating mode using the gyroscope

Fig. 7.3: System overview of MCU implementation of C-HAR system in *matched axis* operating mode using the accelerometer, and the gyroscope.

7.4.2 Feature Extraction and Classification

After loading the $\xi[m]$ measurements for each axis onto the MCU, we extract the following fifteen features:

- mean
- standard deviation
- median
- absolute largest value
- interquartile range

for

- x-axis
- y-axis
- z-axis

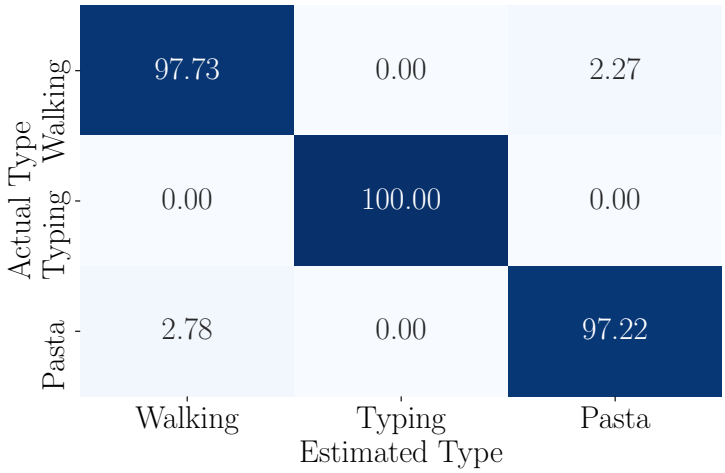
We confirm that the features extracted by the MCU are the same as the features extracted in software by outputting the relevant features to the Teensy IDE's serial monitor and comparing them to the feature values generated in software.

The classifier used is a smaller ported version of the RF classifier trained in the system's software implementation. We set the number of trees and the minimum number of samples per leaf parameters of the RF classifier as $\text{n\_estimators} = 100$ and $\text{min\_samples\_leaf} = 3$ respectively, with the other parameters left as default.

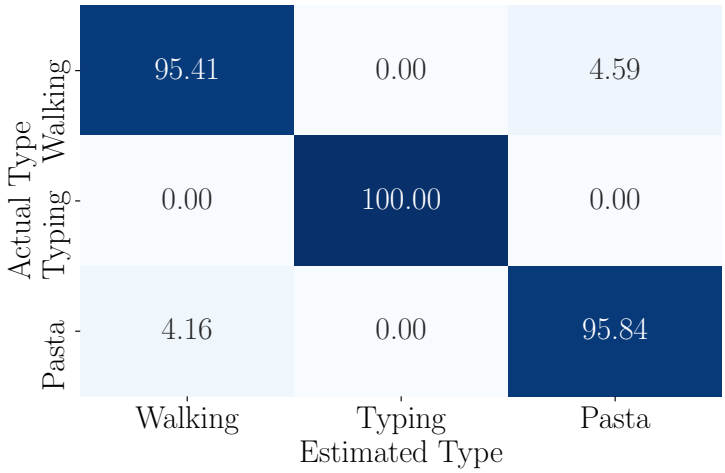
7.4.3 Evaluation and Discussion

We obtain an accuracy of 98.3% for the accelerometer-based system and 97.1% for the gyroscope-based system, and show the corresponding confusion matrices in Figure 6.7. These results are comparable to the simulation accuracies of 98.5% for the accelerometer-based system and 97.6% for the gyroscope-based system obtained in Section 6.4.2. Despite the reduction in model size from 1000 trees to 100 trees, the system's accuracy performance remains stable. This would suggest that the sensors and signals used in our application are inherently well suited to the C-HAR approach, given the high accuracy obtained in both the software and MCU implementations of the system. It is important to note, however, that the MCU implementation of

the C-HAR uses a reduced dataset, and that system performance may vary if confronted with a different dataset or the entire original dataset.



(a) Accelerometer-based MCU implementation results



(b) Gyroscope-based MCU implementation results

Fig. 7.4: Confusion matrices for accelerometer-based MCU implementation results and gyroscope-based MCU implementation results. Average accuracy is 98.3% and 97.1%, respectively.

7.5 Results and Discussion

From the results, we can deduce that the model size has the biggest single effect on the performance of the MCU implementations of our proposed sensing architectures. Indeed, given that the extracted features are the

same in both the simulation and MCU implementations of our systems, it is clear that the discrepancies in results are due to the differences between the original and ported models.

The difference in performance between the C-AVDI MCU implementation and the C-HAR MCU implementation in relation to their simulation-based systems can be explained by the nature of the original systems themselves. C-HAR uses three-dimensional signals acquired by wearable sensors in environments where the main potential source of noise is electronic noise produced by the device circuitry. This leads to a relatively high Signal-to-Noise Ratio (SNR), which in turn makes classification more straightforward and performable using a smaller model. In contrast, C-AVDI uses a single-dimensional signal obtained by a microphone in environments with significant and constant background noise, whose nature and amplitude vary greatly depending on the weather conditions. This leads to a significantly lower SNR, which in turn makes classification more difficult, requiring the use of a larger, more complex model.

We saw in Section 7.1.2 that there is an increasing amount of research focussing on developing new approaches to implementing complex classifiers on resource-constrained MCUs. The performance of the C-AVDI system's MCU implementation could conceivably be improved by using these techniques rather than just using a standard C porting library. The creation of an MCU-specific classifier which is tailored to accepting compressive measurements as input is left as a potential avenue for future work.

7.6 Summary

In this chapter, we covered the process of implementing our C-AVDI and C-HAR systems on an MCU. Both systems were implemented on the same MCU and the respective classifiers were ported using the same library, which in both cases required the classifier model to be reduced in size. While this reduction in size had a negligible effect on the performance of the MCU implementation of C-HAR, it had a significant impact on the MCU implementation of C-AVDI, reducing the classification accuracy of the system from 80.1% to 71.8%.

8 Conclusion and Future Work

8.1 Conclusion

In this thesis, we presented a lightweight compressive measurement-based classification architecture capable of identifying a range of input signals at sub-Nyquist rates, while presenting accuracy comparable to currently existing sensing-to-classification systems, all while running on a resource-constrained device. By reducing the sample rate used to acquire the various signals encountered by different sensing architectures, we aimed to solve the three-pronged problem of energy usage, system computational runtime, and model size at the source.

This was achieved first and foremost by developing a compressive-measurement based sensing framework capable of simultaneously filtering and sampling signals at sub-Nyquist rates. We designed a Markov chain-based signal generation architecture to create spectrally tailored bipolar spreading sequences which actively attenuate or amplify specific frequency content in an input signal, and incorporated it into a random demodulator architecture (RD) to obtain a reduced set of filtered compressive measurements for use in the subsequent classification stage.

This architecture was then used in two different sensing scenarios to demonstrate its viability as an application-agnostic sub-Nyquist sensing framework. We first presented a compressive measurement-based acoustic vehicle detection and identification system (C-AVDI), which classified the sounds of passing vehicles under a variety of weather conditions. We obtained an accuracy comparable to that of a baseline acoustic vehicle detection and identification (AVDI) system, at a significantly reduced sampling rate. We

then presented a compressive measurement-based human activity recognition system (C-HAR), which classified a wide range of daily human activities using data obtained from the sensor readings of a smartwatch-embedded accelerometer and gyroscope. The system obtained accuracy comparable to benchmark and other state-of-the-art lightweight human activity recognition (HAR) systems, at a similar sample rate and computational cost to these existing systems.

Finally, we provided an insight into the feasibility and viability of implementing both C-AVDI and C-HAR on a microcontroller (MCU). This was achieved by porting the respective trained classifiers from the software environment to a resource constrained MCU device. The limited memory meant that the ported classifiers were smaller in size than the original versions, and while this difference had a negligible effect on the performance of C-HAR, it had a significant negative impact on the performance of C-HAR.

Overall, we believe that our results illustrate the potential of our proposed architecture as an application-agnostic, low-cost, low-complexity sensing to classification framework. We believe that our approach to low-rate sensing and reduced dimensionality signal processing can play a significant role in reducing the financial and environmental costs of machine learning. By helping to democratise the use and access to these technologies, we hope to drive innovation, research, and ultimately technological progress. In the subsequent sections, we present a more detailed description of the contributions and limitations of our work, as well as any potential directions for future research

8.1.1 Contributions

In the course of the research presented in this thesis, we made a number of contributions to help advance the field of lightweight sensing and classification. These contributions can be grouped into the following categories:

- *Theoretical:* in terms of theoretical contributions, in Chapter 4 we expanded and improved upon the initial Markov chain-based bipolar sequence generation method by introducing dual Markov chain-generated sequences, as well as demonstrating and developing the concept of

using signal matching as a way to attenuate or amplify input signal frequency content. We used these new developments in Chapter 5 to create the final bipolar sequence used to evaluate our C-AVDI system. In both Chapter 5 and Chapter 6, we observed that the minimum number of M measurements required for optimal classification accuracy is noticeably lower than the minimum number of M measurements required for signal reconstruction, as seen in Chapter 4. This suggests that when using compressive measurements for classification rather than reconstruction, it is possible to relax the lower bound on the number of measurements dictated by Equation 2.9, further reducing the system's sampling rate. It is important to note that this conclusion is based on observed results obtained in our simulations, and that further experiments are required before we can definitively draw any conclusions, or make any further assertions regarding this matter.

- *Practical:* in terms of practical contributions, we conceptualised, designed, and evaluated two sub-Nyquist compressive measurement-based sensing-to-classification systems. In Chapter 5, we implemented our proposed approach in an AVDI system, and in Chapter 6 we implemented our proposed approach in a HAR system. Both these systems presented a certain number of advantages when compared to existing approaches, helping to advance the state-of-the-art in their respective fields. We also presented in Chapter 7 an initial attempt at implementing the aforementioned C-AVDI and C-HAR systems on MCUs. While this first attempt is successful in porting C-HAR to a smaller device, it is less successful with C-AVDI, whose MCU implementation shows a drop in accuracy performance compared to its software implementation. Despite this, the results are encouraging, and the accuracy performance is sufficient to prove the viability of MCU implementations of our proposed sub-Nyquist compressive measurement-based architectures.

- *Data:* in terms of data-related contributions, we created a vehicle sound dataset by recording the sounds of vehicles passing on a two-lane, two-way road under a variety of weather conditions over the course of 14 days. The recordings were labelled and segmented before being added

to the dataset and used in Chapter 5. Further details regarding the dataset creation process can be found in Section 5.2.

8.1.2 Limitations and Future Work

We encountered a number of limitations during our research. We believe that resolving these limitations could present a few potential avenues for future work. Both are listed below.

- *Spreading sequence limitations:* while the conceptualisation and implementation of dual Markov chain-generated spreading sequences is one of the key contributions of our research, there is still a limit on the length of the chains themselves. Indeed, the frequency spectrum of a spreading sequence generated by a more than 4-state Markov-chain presents significant ripple across the entire spectrum, rendering it unusable in our proposed systems. This in turn places a limit on the range of different usable spreading sequences that can be generated and used to filter and demodulate input signals. The design of an alternative spreading sequence generation method capable of producing increasingly complex, but still usable, sequences is left as a subject of future research.
- *Adverse weather conditions:* a significant limitation of the proposed C-AVDI system is the decrease in accuracy performance observed in adverse weather conditions, particularly in windy conditions where the accuracy drops significantly. Improving the system's performance in these conditions will have a significant effect on the overall average operating accuracy of the system. This limitation can be mitigated in future work by incorporating an additional passive or active noise cancellation stage as seen in our previous work [79] [80], by implementing CSP-based interference removal techniques as presented in [65], or by designing more complex bipolar sequences for more precise filtering during signal acquisition.
- *Microcontroller implementation of classifiers:* The limited amount of available memory on the MCU restricts the size of the random forest (RF) classifiers used in the estimation process. While this is not a

problem in C-HAR, the reduction in the number of trees in the ported C-AVDI RF classifier has a negative impact on the system's performance. While there are many inherent factors that explain the difference in minimum required model size between C-AVDI and C-HAR, such as the nature of the signals, sensors, and typical resulting signal-to-noise ratios present in both applications, the most effective and efficient way to improve the MCU performance of a compressive measurement-based system is to modify the way in which the classifiers are ported from software to MCU device. Indeed, in Chapter 7, the classifiers are ported using a general library which simply converts a trained *scikit-learn* model to an MCU-oriented language, which while straightforward and effective in more typical situations, does not offer any type of model size optimisation. In our work, we focussed first and foremost on creating an entire lightweight sensing architecture, and the comparatively wide scope of our research meant that there were certain components of our proposed framework that received less attention than others. Future work could include the creation of a novel approach to implementing compressive measurement-oriented classifiers on MCUs, which leverages the inherent dimensionality reduction properties of CS to optimise both model size and performance.

Bibliography

[1]Melvin M. Vopson. *The world's data explained: how much we're producing and where it's all stored*. https://theconversation.com/the-worlds-data-explained-how-much-were-producing-and-where-its-all-stored-159964. Published: 2021-5-4, Accessed: 2022-1-24 (cit. on p. 1).

[2]Dario Amodei, Danny Hernandez, Girish Sastry, et al. *AI and Compute*. https://openai.com/blog/ai-and-compute/. Published: 2018-5-16, Accessed: 2020-9-17. 2018 (cit. on p. 1).

[3]Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. "Green AI". In: *Commun. ACM* 63.12 (Nov. 2020), 54–63 (cit. on pp. 1, 3).

[4]Emma Strubell, Ananya Ganesh, and Andrew McCallum. "Energy and Policy Considerations for Deep Learning in NLP". In: *Proc. the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, July 2019, pp. 3645–3650 (cit. on p. 2).

[5]Tom B. Brown, Benjamin Mann, Nick Ryder, et al. "Language Models are Few-Shot Learners". In: *Computing Research Repository (CoRR)* abs/2005.14165 (July 2020). arXiv: 2005.14165 (cit. on p. 2).

[6]Chuan Li. *OpenAI's GPT-3 language model: a technical overview*. https://lambdalabs.com/blog/demystifying-gpt-3/. Published: 2020-6-3, Accessed: 2022-5-23 (cit. on p. 2).

[7]Shaden Smith, Mostofa Patwary, Brandon Norick, et al. "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model". In: *Computing Research Repository (CoRR)* abs/2201.11990 (Feb. 2022). arXiv: 2201.11990 (cit. on p. 2).

[8]Will Knight. *AI's smarts now come with a big price tag*. https://www.wired.com/story/ai-smarts-big-price-tag/. Published: 2021-10-14, Accessed: 2021-12-29 (cit. on p. 2).

[9]Ana Lima, Tim Hahn, Luc Evers, et al. "Feasibility of large-scale deployment of multiple wearable sensors in Parkinson's disease". In: *PLOS ONE* 12 (Dec. 2017), e0189161 (cit. on pp. 3, 18, 78).

[10]Aosen Wang, Feng Lin, Zhanpeng Jin, and Wenyao Xu. “A Configurable Energy-Efficient Compressed Sensing Architecture With Its Application on Body Sensor Networks”. In: *IEEE Transactions on Industrial Informatics* 12 (Jan. 2015), pp. 1–1 (cit. on pp. 5, 79).

[11]Peter Prince, Andrew Hill, Evelyn Piña Covarrubias, et al. “Deploying Acoustic Detection Algorithms on Low-Cost, Open-Source Acoustic Sensors for Environmental Monitoring”. In: *Sensors* 19.3 (Jan. 2019) (cit. on pp. 5, 17, 60, 104, 105).

[12]Ganapati Bhat, Nicholas Tran, Holly Shill, and Umit Y. Ogras. “w-HAR: An Activity Recognition Dataset and Framework Using Low-Power Wearable Devices”. In: *Sensors* 20.18 (Sept. 2020) (cit. on p. 5).

[13]Antoine Back, Paul Chollet, Olivier Fercoq, and Patricia Desgreys. “Power-aware feature selection for optimized Analog-to-Feature converter”. In: *Microelectronics Journal* 122 (Apr. 2022), p. 105386 (cit. on pp. 5, 6, 18, 79, 101).

[14]Shigemi Ishida, Jumpei Kajimura, Masato Uchino, Shigeaki Tagashira, and Akira Fukuda. “SAVeD: Acoustic Vehicle Detector with Speed Estimation capable of Sequential Vehicle Detection”. In: *Proc. IEEE Conf. Intelligent Transportation Systems (ITSC)*. Nov. 2018, pp. 906–912 (cit. on pp. 7, 16, 22, 30).

[15]David L. Donoho. “Compressed sensing”. In: *IEEE Transactions on Information Theory* 52.4 (Apr. 2006), pp. 1289–1306 (cit. on pp. 10, 11).

[16]Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information”. In: *IEEE Transactions on Information Theory* 52.2 (Feb. 2006), 489–509 (cit. on p. 11).

[17]Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. “Atomic Decomposition by Basis Pursuit”. In: *SIAM J. Sci. Comput.* 20.1 (Jan. 1998), 33–61 (cit. on p. 13).

[18]Joel A. Tropp and Anna C. Gilbert. “Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit”. In: *IEEE Transactions on Information Theory* 53.12 (Dec. 2007), pp. 4655–4666 (cit. on p. 13).

[19]Emmanuel J. Candès and Terence Tao. “Decoding by linear programming”. In: *IEEE Transactions on Information Theory* 51.12 (Dec. 2005), pp. 4203–4215 (cit. on pp. 13, 42).

[20]Emmanuel J. Candès and Michael B. Wakin. “An Introduction To Compressive Sampling”. In: *IEEE Signal Process. Mag.* 25.2 (Mar. 2008), pp. 21–30 (cit. on p. 13).

[21]Meenu Rani, Sanjay B. Dhok, and Raghavendra B. Deshmukh. “A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications”. In: *IEEE Access* 6 (Jan. 2018), pp. 4875–4894 (cit. on p. 13).

[22]Sami Kirolos, Jason N. Laska, Michael Wakin, et al. “Analog-to-Information Conversion via Random Demodulation”. In: *Proc. 2006 IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*. Oct. 2006, pp. 71–74 (cit. on p. 13).

[23]Jason N. Laska, Sami Kirolos, Marco F. Duarte, et al. “Theory and Implementation of an Analog-to-Information Converter using Random Demodulation”. In: *Proc. 2007 IEEE International Symposium on Circuits and Systems*. May 2007, pp. 1959–1962 (cit. on p. 13).

[24]Joel A. Tropp, Jason N. Laska, Marco F. Duarte, Justin K. Romberg, and Richard G. Baraniuk. “Beyond Nyquist: Efficient Sampling of Sparse Bandlimited Signals”. In: *IEEE Transactions on Information Theory* 56.1 (Jan. 2010), 520–544 (cit. on pp. 13, 15).

[25]Moshe Mishali, Yonina C. Eldar, Oleg Dounaevsky, and Eli Shoshan. “Xampling: Analog to digital at sub-Nyquist rates”. In: *IET Circuits Devices Syst.* 5.1 (2011), pp. 8–20 (cit. on pp. 13, 39, 41).

[26]Tanbir Haque, Rabia Tugce Yazicigil, Kyle Jung-Lin Pan, John Wright, and Peter R. Kinget. “Theory and Design of a Quadrature Analog-to-Information Converter for Energy-Efficient Wideband Spectrum Sensing”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 62.2 (Feb. 2015), pp. 527–535 (cit. on p. 13).

[27]Michael A. Lexa, Mike E. Davies, and John S. Thompson. “Reconciling Compressive Sampling Systems for Spectrally Sparse Continuous-Time Signals”. In: *IEEE Transactions on Signal Processing* 60.1 (Jan. 2012), pp. 155–171 (cit. on p. 13).

[28]Richard Baraniuk, Jason N. Laska, Mark A. Davenport, et al. *An Introduction to Compressive Sensing*. OpenStax, Apr. 2011 (cit. on p. 15).

[29]Guoliang Mo and Sanyuan Zhang. “Vehicles detection in Traffic Flow”. In: *Proc. 2010 Sixth International Conference on Natural Computation*. Vol. 2. Aug. 2010, pp. 751–754 (cit. on p. 15).

[30]Amir Mukhtar, Likun Xia, and Tong Boon Tang. “Vehicle Detection Techniques for Collision Avoidance Systems: A Review”. In: *IEEE Transactions on Intelligent Transportation Systems* 16 (Oct. 2015), pp. 1–21 (cit. on p. 15).

[31]Marcin Słomiany, Piotr Gemza, Filip Jędrzejczyk, Mateusz Maciaś, and Jakub Główka. “System for Detection of Vehicles in Multiple Video Streams in Road Infrastructure Monitoring”. In: *Proc. Automation 2020: Towards Industry of the Future*. Springer International Publishing, Mar. 2020, pp. 157–169 (cit. on p. 15).

[32]Bai Hongliang and Liu Changping. “A Hybrid License Plate Extraction Method Baed On Edge Statistics and Morphology”. In: *Proc. Int. Conf. Pattern Recognition (ICPR)*. Vol. 2. Aug. 2004, pp. 831–834 (cit. on pp. 16, 20).

[33]Tamas Lovas, Charles Toth, and Arpad Barsi. “Model-based vehicle detection from lidar data”. In: *Proc. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. Jan. 2004 (cit. on p. 16).

[34]Bence Major, Daniel Fontijne, Amin Ansari, et al. “Vehicle Detection With Automotive Radar Using Deep Learning on Range-Azimuth-Doppler Tensors”. In: *Proc. 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. Oct. 2019, pp. 924–932 (cit. on p. 16).

[35]Shallaw Sheik Mohammed Ali, Boby George, Lelitha Vanajakshi, Vinod Jayashankar, and Vinod Jagadeesh Kumar. “A multiple loop vehicle detection system for heterogeneous and lane-less traffic”. In: *Proc. 2011 IEEE International Instrumentation and Measurement Technology Conference*. Dec. 2011, pp. 1–5 (cit. on p. 16).

[36]Honghui Dong, Xuzhao Wang, Chao Zhang, et al. “Improved Robust Vehicle Detection and Identification Based on Single Magnetic Sensor”. In: *IEEE Access* 6 (Jan. 2018), pp. 5247–5255 (cit. on p. 16).

[37]Myounggyu Won, Sayan Sahu, and Kyung-Joon Park. “DeepWiTraffic: Low Cost WiFi-Based Traffic Monitoring System Using Deep Learning”. In: *Pro. 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. Nov. 2019, pp. 476–484 (cit. on p. 16).

[38]Barbara Barbagli, Luca Bencini, Iacopo Magrini, Gianfranco Manes, and Antonio Manes. “A real-time traffic monitoring based on wireless sensor network technologies”. In: *Proc. 2011 7th International Wireless Communications and Mobile Computing Conference*. July 2011, pp. 820–825 (cit. on p. 16).

[39]Billy Dawton, Shigemi Ishida, Yuki Hori, Masato Uchino, and Yutaka Arakawa. “Proposal for a Compressive Measurement-Based Acoustic Vehicle Detection and Identification System”. In: *Proc. IEEE Vehicular Technology Conf.* Nov. 2020, pp. 1–6 (cit. on pp. 16, 60, 61, 67).

[40]Billy Dawton, Shigemi Ishida, Yuki Hori, et al. “Initial Evaluation of Vehicle Type Identification using Roadside Stereo Microphones”. In: *Proc. IEEE Sensors and Applications Symp. (SAS)*. Mar. 2020, pp. 1–6 (cit. on pp. 16, 19, 58–60, 72, 73).

[41]Enrique Alexandre, Lucas Cuadra, Sancho Salcedo-Sanz, Álvaro Pastor-Sánchez, and Carlos. Casanova-Mateo. “Hybridizing Extreme Learning Machines and Genetic Algorithms to Select Acoustic Features in Vehicle Classification Applications”. In: *Neurocomput.* 152.C (Mar. 2015), 58–68 (cit. on pp. 16, 59, 60).

[42]Billy Dawton, Shigemi Ishida, and Yutaka Arakawa. “C-AVDI: Compressive Measurement-Based Acoustic Vehicle Detection and Identification”. In: *IEEE Access* 9 (Dec. 2021), pp. 159457–159474 (cit. on p. 16).

[43]Hüseyin Göksu. “Vehicle speed measurement by on-board acoustic signal processing”. In: *Measurement and Control* 51.5-6 (July 2018), pp. 138–149 (cit. on pp. 16, 58).

[44]Ahmad Aljaafreh and Liang Dong. “An Evaluation of Feature Extraction Methods for Vehicle Classification Based On Acoustic Signals”. In: *Proc. IEEE Int. Conf. on Networking, Sensing, and Control*. Apr. 2010, pp. 570–575 (cit. on pp. 16, 20, 59, 60).

[45]Justin Sunu, Allon G. Percus, and Blake Hunter. “Unsupervised Vehicle Recognition Using Incremental Reseeding of Acoustic Signatures”. In: *Foundations of Intelligent Systems, Proc. 17th International Symposium (ISMIS)*. Cham: Springer International Publishing, Feb. 2018, pp. 151–160 (cit. on p. 16).

[46]Shigemi Ishida, Kohei Mimura, Liu Song, Shigeaki Tagashira, and Akira Fukuda. “Design and Initial Evaluation of Acoustic Vehicle Count System utilizing Dynamic Time Warping”. In: *IPSJ SIG Tech. Reports*. (ITS-64-6), Mar. 2016, pp. 1–6 (cit. on p. 17).

[47]Kazuo Kubo, Chengyu Li, Shigemi Ishida, Shigeaki Tagashira, and Akira Fukuda. “Design of Ultra Low Power Vehicle Detector utilizing Discrete Wavelet Transform”. In: *Proc. ITS AP Forum*. May 2018, pp. 1052–1063 (cit. on pp. 17, 59, 60, 104).

[48]Rex Liu, Albara Ramli, Huanle Zhang, Erik Henricson, and Xin Liu. “An Overview of Human Activity Recognition Using Wearable Sensors: Healthcare and Artificial Intelligence”. In: *Proc. 6th International Conference Internet of Things (ICIOT)*. Springer-Verlag, Jan. 2022, pp. 1–14 (cit. on p. 17).

[49] Sangavi Sangavi and Mohammed B.A. Hashim. “Human Activity Recognition for Ambient Assisted Living”. In: *Proc. 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*. Mar. 2019, pp. 1–4 (cit. on p. 17).

[50] Wei Niu, Jiao Long, Dan Han, and Yuan-Fang Wang. “Human activity detection and recognition for video surveillance”. In: *Proc. 2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*. Vol. 1. June 2004, 719–722 Vol.1 (cit. on p. 17).

[51] Gary M. Weiss, Kenichi Yoneda, and Thaier Hayajneh. “Smartphone and Smartwatch-Based Biometrics Using Activities of Daily Living”. In: *IEEE Access* 7 (Sept. 2019), pp. 133190–133202 (cit. on pp. 17, 35, 36, 81).

[52] Kyaw Kyaw Htike, Othman O. Khalifa, Huda Adibah Mohd Ramli, and Mohammad A. M. Abushariah. “Human activity recognition for video surveillance using sequences of postures”. In: *Proc. Third International Conference on e-Technologies and Networks for Development (ICeND2014)*. Apr. 2014, pp. 79–82 (cit. on pp. 17, 34, 37).

[53] Shangyue Zhu, Junhong Xu, Hanqing Guo, et al. “Indoor Human Activity Recognition Based on Ambient Radar with Signal Processing and Machine Learning”. In: *Proc. 2018 IEEE International Conference on Communications (ICC)*. May 2018, pp. 1–6 (cit. on pp. 17, 34, 37).

[54] Sheheryar Arshad, Chunhai Feng, Yonghe Liu, et al. “Wi-chase: A WiFi based human activity recognition system for sensorless environments”. In: *Proc. 2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. 2017, pp. 1–6 (cit. on pp. 17, 34, 37).

[55] Yugo Nakamura, Yutaka Arakawa, Takuya Kanehira, Masashi Fujiwara, and Keiichi Yasumoto. “SenStick: Comprehensive Sensing Platform with an Ultra Tiny All-In-One Sensor Board for IoT Research”. In: *Journal of Sensors* 2017 (Oct. 2017), pp. 1–16 (cit. on pp. 18, 34).

[56] Hristijan Gjoreski, Mathias Ciliberto, Lin Wang, et al. “The University of Sussex-Huawei Locomotion and Transportation Dataset for Multimodal Analytics With Mobile Devices”. In: *IEEE Access* 6 (July 2018), pp. 42592–42604 (cit. on pp. 18, 34).

[57] Eloise G. Zimbelman and Robert F. Keefe. “Development and validation of smartwatch-based activity recognition models for rigging crew workers on cable logging operations”. In: *PLOS ONE* 16.5 (May 2021), pp. 1–25 (cit. on pp. 18, 34, 37).

[58] Ganapati Bhat, Yigit Tuncel, Sizhe An, Hyung Gyu Lee, and Umit Y. Ogras. “An Ultra-Low Energy Human Activity Recognition Accelerator for Wearable Health Applications”. In: *ACM Trans. Embed. Comput. Syst.* 18.5s (Oct. 2019) (cit. on pp. 18, 105).

[59] Ryan P. Avery, Yinhai Wang, and Scott Rutherford. “Length-Based Vehicle Classification Using Images from Uncalibrated Video Cameras”. In: *Proc. IEEE Conf. Intelligent Transportation Systems (ITSC)*. Oct. 2004, pp. 1–6 (cit. on p. 20).

[60] Zhang Changjun and Chen Yuzong. “The Research of Vehicle Classification Using SVM and KNN in a Ramp”. In: *Proc. Int. Forum on Computer Science-Technology and Applications*. Dec. 2009, pp. 391–394 (cit. on p. 20).

[61] Seung S. Yang, Yoon G. Kim, and Hongsik Choi. “Vehicle Identification using Wireless Sensor Networks”. In: *Proc. IEEE SoutheastCon*. Mar. 2007, pp. 41–46 (cit. on p. 20).

[62] Martin A. Fischler and Robert C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Commun. ACM* 24.6 (June 1981), 381–395 (cit. on p. 25).

[63] Serkan Ballı, Ensar Sağbaş, and Musa Peker. “Human activity recognition from smart watch sensor data using a hybrid of principal component analysis and random forest algorithm”. In: *Measurement and Control* 52 (Nov. 2019) (cit. on pp. 35, 37, 78).

[64] Bolu Oluwalade., Sunil Neela., Judy Wawira., Tobiloba Adejumo., and Saptarshi Purkayastha. “Human Activity Recognition using Deep Learning Models on Smartphones and Smartwatches Sensor Data”. In: *Proc. 14th International Joint Conference on Biomedical Engineering Systems and Technologies - HEALTHINF,* INSTICC. SciTePress, Jan. 2021, pp. 645–650 (cit. on pp. 35, 37, 78).

[65] Mark A. Davenport, Petros T. Boufounos, Michael B. Wakin, and Richard G. Baraniuk. “Signal Processing With Compressive Measurements”. In: *IEEE Journal of Selected Topics in Signal Processing* 4.2 (Apr. 2010), pp. 445–460 (cit. on pp. 39, 40, 74, 118).

[66] Thakshila Wimalajeewa and P.K. Varshney. “Compressive Sensing Based Classification in the Presence of Intra- and Inter- Signal Correlation”. In: *IEEE Signal Processing Letters* 25 (July 2018), pp. 1–1 (cit. on p. 39).

[67] Xinming Liu, Emre Gönültaş, and Christoph Studer. “Analog-to-Feature (A2F) Conversion for Audio-Event Classification”. In: *Proc. 2018 26th European Signal Processing Conference (EUSIPCO)*. Sept. 2018, pp. 2275–2279 (cit. on p. 39).

[68]Moshe Mishali, Asaf Elron, and Yonina C. Eldar. “Sub-Nyquist processing with the modulated wideband converter”. In: *Proc. 2010 IEEE International Conference on Acoustics, Speech and Signal Processing*. Mar. 2010, pp. 3626–3629 (cit. on p. 40).

[69]Andrew Harms, Waheed U. Bajwa, and A. Robert Calderbank. “A Constrained Random Demodulator for Sub-Nyquist Sampling”. In: *IEEE Trans. Signal Process.* 61.3 (Feb. 2013), pp. 707–723 (cit. on pp. 40–42, 66).

[70]Dian Mo and Marco F. Duarte. “Design of spectrally shaped binary sequences via randomized convex relaxations”. In: *Proc. 2015 49th Asilomar Conference on Signals, Systems and Computers*. Nov. 2015, pp. 164–168 (cit. on p. 41).

[71]Andrew Harms, Waheed U. Bajwa, and A. Robert Calderbank. “Shaping the power spectra of bipolar sequences with application to sub-Nyquist sampling”. In: *Proc. 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP*. IEEE, Dec. 2013, pp. 236–239 (cit. on pp. 43, 66).

[72]Ewout van den Berg and Michael P. Friedlander. *SPGL1: A solver for large-scale sparse reconstruction*. `https://friedlander.io/spgl1`. Dec. 2019 (cit. on p. 47).

[73]Hüseyin Göksu. “Engine Speed-Independent Acoustic Signature for Vehicles”. In: *Measurement and Control* 51.3–4 (Apr. 2018), pp. 94–103 (cit. on pp. 58–60).

[74]Jobin George, Leena Mary, and Riyas K S. “Vehicle detection and classification from acoustic signal using ANN and KNN”. In: *Proc. 2013 International Conference on Control Communication and Computing (ICCC)*. Dec. 2013, pp. 436–439 (cit. on pp. 58–60).

[75]Yannick Schulz, Avinash Kini Mattar, Thomas M. Hehn, and Julian F. P. Kooij. “Hearing What You Cannot See: Acoustic Vehicle Detection Around Corners”. In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 2587–2594 (cit. on pp. 59, 60).

[76]Haoze Chen and Zhijie Zhang. “Hybrid neural network based on novel audio feature for vehicle type identification”. In: *Sci Rep* 11 (Apr. 2021) (cit. on pp. 59, 60).

[77]Luca Becker, Alexandru Nelus, Johannes Gauer, Lars Rudolph, and Rainer Martin. “Audio Feature Extraction for Vehicle Engine Noise Classification”. In: *Proc. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. May 2020, pp. 711–715 (cit. on pp. 59, 60).

[78]Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (Feb. 2011), pp. 2825–2830 (cit. on pp. 68, 94).

[79]Masato Uchino, Shigemi Ishida, Kazuo Kubo, Shigeaki Tagashira, and Akira Fukuda. “Initial Design of Acoustic Vehicle Detector with Wind Noise Suppressor”. In: *Proc. Int. Workshop on Pervasive Computing for Vehicular Systems (PerVehicle)*. Mar. 2019, pp. 814–819 (cit. on pp. 74, 118).

[80]Shigemi Ishida, Masato Uchino, Chengyu Li, Shigeaki Tagashira, and Akira Fukuda. “Design of Acoustic Vehicle Detector with Steady-Noise Suppression”. In: *Proc. IEEE Conf. Intelligent Transportation Systems (ITSC)*. Oct. 2019 (cit. on pp. 74, 118).

[81]Mordor Intelligence. *Global Smartwatch Market - Growth, Trends, COVID-19 Impact, and Forecasts (2022 - 2027)*. `https://www.mordorintelligence.com/industry-reports/smartwatch-market`. Published: 2022, Accessed: 2022-5-17. 20022 (cit. on p. 77).

[82]Anneli Ozanne, Dongni Buvarp, Ulla Graneheim, et al. “Wearables in epilepsy and Parkinson’s disease-A focus group study”. In: *Acta neurologica Scandinavica* 137 (July 2017) (cit. on p. 78).

[83]Mohammed Shoaib, Kyong Ho Lee, Niraj K. Jha, and Naveen Verma. “A 0.6–107 μW Energy-Scalable Processor for Directly Analyzing Compressively-Sensed EEG”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 61.4 (Jan. 2014), pp. 1105–1118 (cit. on pp. 79, 80, 101).

[84]Md Mohaimenuzzaman, Christoph Bergmeir, Ian Thomas West, and Bernd Meyer. “Environmental Sound Classification on the Edge: A Pipeline for Deep Acoustic Networks on Extremely Resource-Constrained Devices”. In: *ArXiv* abs/2103.03483 (Apr. 2021) (cit. on p. 105).

[85]Ilana Stolovas, Santiago Suárez, Diego Pereyra, Francisco De Izaguirre, and Varinia Cabrera. “Human activity recognition using machine learning techniques in a low-resource embedded system”. In: *Proc. 2021 IEEE URUCON*. Nov. 2021, pp. 263–267 (cit. on p. 105).

[86]Pierre-Emmanuel Novac, Andrea Castagnetti, Adrien Russo, et al. “Toward unsupervised Human Activity Recognition on Microcontroller Units”. In: *Proc. 2020 23rd Euromicro Conference on Digital System Design (DSD)*. Aug. 2020, pp. 542–550 (cit. on p. 105).

[87]Wenbo Xu, Yupeng Cui, Yue Wang, Siye Wang, and Jiaru Lin. “A hardware implementation of random demodulation analog-to-information converter”. In: *IEICE Electronics Express* 13.16 (Aug. 2016), pp. 20160465–20160465 (cit. on p. 106).

[88]Jon Nordby. *emlearn: Machine Learning inference engine for Microcontrollers and Embedded Devices*. `https://doi.org/10.5281/zenodo.2589394`. Mar. 2019 (cit. on p. 107).